1979-03

Seaguard: a general purpose antiship missile defense computer simulation.

Callahan, Alexander Joseph

Monterey, California : Naval Postgraduate School

http://hdl.handle.net/10945/18860

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SEAGUARD
A GENERAL PURPOSE ANTISHIP
MISSILE DEFENSE COMPUTER SIMULATION

by

Alexander Joseph Callahan, Jr.

March 1979

Thesis Advisor: E.F. Roland

Approved for public release; distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

SEAGUARD, A General Purpose Antiship Missile Defense Computer Simulation

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis
March 1979

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Alexander Joseph Callahan, Jr.

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

March 1979

13. NUMBER OF PAGES

81

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Antiship Missile Defense Simulation
Computer Simulation
Missile Defense
Naval Computer Simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SEAGUARD is an event store computer simulation of the antiship missile defense of a naval task force using surface to air and air to air missiles. It simulates the interaction between the antiship missiles and the defensive missiles of a task force at sea in a non-jamming environment. Using SEAGUARD, the military analyst can measure the relative effectiveness of alternative defensive dispositions or alternative forces. SEAGUARD was designed to be

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

#20. ABSTRACT (CONT'D)

general in order to facilitate the analysis and avoid the use of a large or extremely detailed data base.

SEAGUARD
A General Purpose Antiship Missile Defense Computer Simulation

by

Alexander Joseph Callahan, Jr.
Lieutenant, United States Navy
B.A., Jersey City State College, 1973

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1979

ABSTRACT

SEAGUARD is an event store computer simulation of the antiship missile defense of a naval task force using surface to air and air to air missiles. It simulates the interaction between the antiship missiles and the defensive missiles of a task force at sea in a non-jamming environment. Using SEAGUARD, the military analyst can measure the relative effectiveness of alternative defensive dispositions or alternative forces. SEAGUARD was designed to be general in order to facilitate the analysis and avoid the use of a large or extremely detailed data base.

TABLE OF CONTENTS

I. INTRODUCTION

A. THE NEED FOR ANTISHIP MISSILE DEFENSE EVALUATION

The United States, in recent years, has had to reduce its number of warships due to increased procurement costs and austere budget constraints. At the same time, the Soviet Union has consistently increased military procurement. This increase has been reflected by additional types and numbers of antiship missiles and a greater number of subsurface, surface, and air platforms which launch them. The increase in the relative strength of the Soviet Navy, the change in Soviet naval strategy to a more offensive posture, and the decrease in the number of United States aircraft carriers provoked the essential need for effective evaluation of antiship missile defense. This need was intensified as the United States Navy evaluated new tactics to protect its capital ships, and evaluated alternative ship building programs.

B. ANTISHIP MISSILE DEFENSE COMPUTER SIMULATIONS

SPEARS (System Performance Evaluation and Requirements Simulation), a large-scale, anti-air-warfare-oriented, computer simulation model, was jointly developed at the Naval Warfare Research Center of Stanford Research Institute under the auspices of the Office of Naval Research, the Office of the Chief of Naval Operations and the Naval Ship Systems

Command.[1] SPEARS is a highly sophisticated simulation requiring a large number of data inputs such as radar configuration of each defensive platform, all antiship missile profiles, detail of communication links between individual units, environmental data, weapon profile data, etc. The data required for this simulation lends itself to a detailed analysis of a specific disposition of particular naval units. It is the opinion of this writer that the task of generating the data base to use this model would be difficult if not impossible on a timely basis and the ability of such a sophisticated model to yield a general result is questionable.

REGAME, a probabilistic event store computer simulation of the interactions between SAM (Surface to Air Missile) systems and aircraft was developed at the Naval Postgraduate School. It was designed to be used as a classroom aid for a graduate course on system simulation. REGAME was initially evaluated to be used as an ASMD (Anti Ship Missile Defense) simulation because of its relatively small computer core and computer time requirements. The structure of REGAME, for example, the ability to play only three missile areas, and the limit to its playing area size rendered it inadequate. REGAME's structural limitations required a relatively large

---

[1]Naval Research Laboratory Report 7958, SPEARS, An AAW Performance Simulation, by D.J. Kaplan (NRL), L.C. Davis (NRL), M.E.B. Owens (NRL), O.F. Forsyth (SRI), and J.J. Penick (SRI), p. 1-1, 9 June 1976.

number of computer runs, and a large number of external calculations when a problem with either more than three missile areas or a problem which had an attack area larger than 180 degrees was to be analyzed. REGAME also did not have the capability of playing defensive AAM (Air to Air Missiles).

Various other studies have been conducted on ASMD; however, security restraints of this report do not allow a discussion of them. None of these other studies were appropriate or met the requirements of this thesis. No other unclassified ASMD simulations were located through library search.

C. SEAGUARD, A BETTER ALTERNATIVE

SEAGUARD, as described in this paper, was developed as a Monte Carlo computer simulation which would use a small amount of computer core and time, readily available data, and yield a general result. The computer code of REGAME was used as a basis for SEAGUARD because REGAME was computer core and time efficient. Chapter II explains the structure of REGAME in terms of the playing area, assumptions, offensive and defensive set up, and the game doctrine. The changes required to restructure REGAME into the ASMD simulation, SEAGUARD, form the basis of Chapter III. In the next chapter, SEAGUARD is discussed in detail. This chapter was written so that it could stand alone and be used as a user's guide to SEAGUARD. The definition of a standard set up of a game

and a parametric output analysis of SEAGUARD comprise Chapter V. Finally, Chapter VI serves to review the conclusions from the research and the parametric analysis, and suggests problems that need to be addressed in future studies.

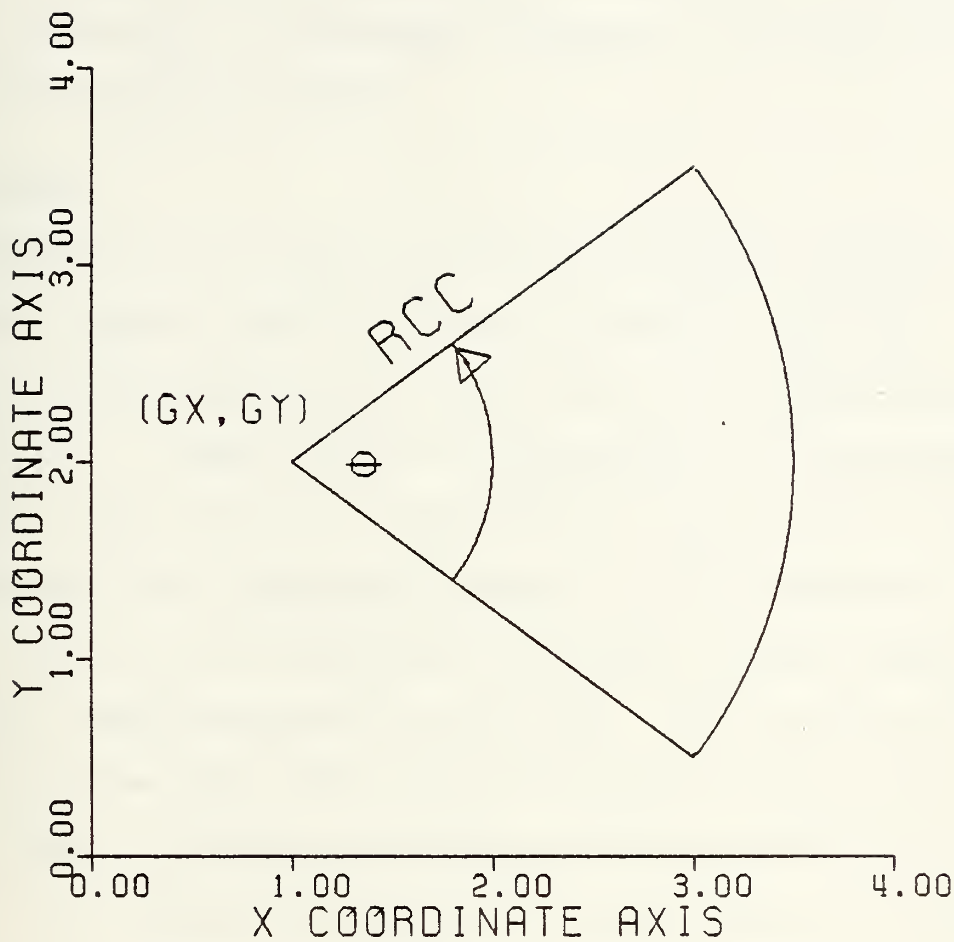
II. REGAME

A. PLAYING AREA

The playing area of REGAME, as illustrated in Figure 1, is a portion of a circle whose radius, RCC, is an input parameter to the simulation. RCC can be as large as 999 miles. The arc length of the playing area is also an input parameter and can be as large as 179 degrees. The bisector of the central playing angle must be parallel to the x coordinate axis of the x,y plane. Defensive missile sites may be placed outside of the playing area; however, only actions that occur within the playing area affect the outcome of the simulation.

B. ASSUMPTIONS

REGAME was written as a small scale, flexible, general purpose model. Certain assumptions had to be made in order to attain the flexibility yet maintain computer storage space at a minimum. All generated incoming aircraft are aimed for the vertex of the central playing angle. The aircraft are assumed to be of the same type; that is, they all have the same distribution of airspeed and altitude. The aircraft are assumed to be observed by all missile areas, subject to the radar horizon and the maximum radar range of the respective missile area. The probability of detection for all radars is assumed to be 1.0 and all fire control solutions are deterministic. The vertex of the playing angle



(GX,GY)	COORDINATES OF CENTER
RCC	RADIUS OF PLAYING AREA
Θ	CENTRAL PLAYING ANGLE

PLAYING AREA

Figure 1

is assumed to represent the bomb release line of the aircraft for their respective targets; therefore, an offensive wave of aircraft is considered successful if at least one aircraft reaches the center.

C. OFFENSE

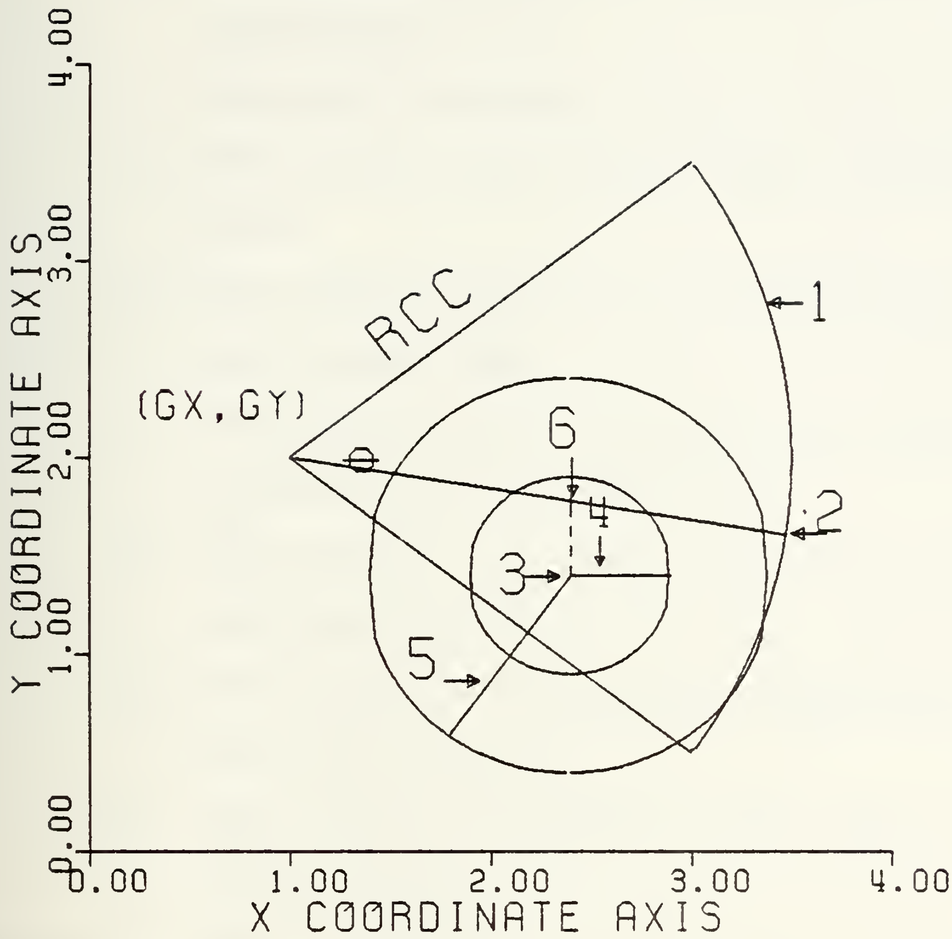
REGAME plays aircraft as offensive weapons against the defensive central point, (GX,GY), of the playing area. The aircraft originate along the outer arc of the playing area, see Figure 2. Their points of origination are distributed uniformly over that arc. The speed of the aircraft, aircraft altitude and time separation between aircraft entering the playing area are all uniformly distributed between upper and lower bounds specified by the user. The aircraft follow a direct course from their point of entry on the outer arc of the playing area to the vertex of the central playing angle. The offense can consist of as many as 20 aircraft which play a passive role in that they only serve as a stimuli for the defensive system simulation.

D. DEFENSE

The defense for REGAME is a set of up to three surface to air missile sites. The input parameters for the defense are as follows:

(1) The number of missile areas in the simulation.

(2) The number of tracking radars assigned to each missile area.



1 ENTRY ARC FOR AIRCRAFT

2 TYPICAL AIRCRAFT FLT. PATH

3 LOCATION OF MISSILE AREA

4 MISSILE MAXIMUM RANGE

5 SEARCH RADAR MAXIMUM RANGE

6 POINT OF CLOSEST APPROACH

PLAYING AREA WITH MISSILE AREA
AND AIRCRAFT FLIGHT PATH

Figure 2

(3) The number of missile launchers assigned to each missile area.

(4) The grid coordinates of each missile area.

(5) The search radar maximum range for each missile area.

(6) The average speed of the defensive missile for each missile area.

(7) The single salvo kill probability for the missiles in each missile area.

(8) The minimum and maximum acquisition time for each missile area.

(9) The minimum and maximum assessment time for each missile area.

(10) The minimum and maximum reload time for each missile area.

Acquisition time is defined as the time required to pass an active target from search radar to fire control radar, and assessment time is defined as the time a fire control radar must remain locked on the target to assess the result of the attempted interception.

The playing area including a typical missile area and aircraft flight path is illustrated in Figure 2. The defensive sites detect and fire at aircraft based on the missile area characteristics provided the aircraft has not passed the closest point of approach to the firing missile site. REGAME is constructed such that either of two missile firing procedures

may be used. These procedures are referred to as uncoordinated and coordinated. The selection of the missile firing procedure is inputed by the user. The uncoordinated missile firing procedure allows all missile areas in the simulation to fire missiles at all aircraft that can possibly be fired upon while the coordinated missile firing procedure allows a missile area to fire at an aircraft only if no other missile area is currently engaging that aircraft.[2]

E. GAME DOCTRINE

REGAME simulates the interactions between surface-to-air missiles and aircraft. This task is accomplished by generating a series of events and associated times. There are five types of events which are as follows:

(1) Fire missile salve.

(2) Missile intercept.

(3) Reload missile launcher.

(4) Free tracking radar.

(5) Change of engagement status.

Each event is scheduled according to the time the event occurs, the type event, the aircraft associated with that event and the missile area associated with that event. As aircraft are generated, the possible missile firing events

---

[2] U.S. Naval Postgraduate School Technical Report 67, A Computer Simulation for the Evaluation of Surface-To-Air Missile Systems in a Clear Environment, by A.F. Andrus, p. 11, June 1966.

associated with that aircraft are scheduled. Each event is checked against certain input parameters before being executed. For example, the firing of a missile at an aircraft is checked against the following:

(1) Radar horizon.

(2) Radar range.

(3) Maximum range of the missile.

(4) Fire control radar availability.

(5) Missile launcher availability.

(6) Missile envelope.

(7) Aircraft engagement status.

New events are generated, as a result of a failure of the defense to kill an aircraft. REGAME simulates the action of the defense against the offense in a series of checks of capability, and availability along with the scheduling and execution of events. Target information is passed among defensive units and shoot-look-shoot firing doctrine is used. REGAME is a Monte Carlo simulation by virtue of the random element used in generating several parameters of the events. For instance, an offensive aircraft is launched with a specified speed. This speed is generated by use of a uniform random number generator and the speed interval specified by the user on input. In the case of the intercept event, the probability of kill p(k), is compared to a uniform (0,1) random number, if the random number is less than or equal to p(k) then the outcome is a kill. The random element is also exhibited in the generation and execution of events.

The event time for the reload event and the free tracking radar event are random numbers. The execution of other events, for example the fire event, are dependent upon the reload time and the change of engagement status; therefore, the execution of the schedule of events takes on a random characteristic.

The result of a single play, or replication, of a Monte Carlo simulation model is determined by the sequence of random numbers obtained from the random number generator. The computations in REGAME that use a Monte Carlo technique are assessment time, acquisition time, reload time, aircraft speed, aircraft separation time, aircraft altitude, entry points of aircraft, and the determination of aircraft kill. A second play of the same game situation with a different sequence of random numbers is likely to produce a different result. The importance of such differences depends on the purpose of the analysis.

A series of replications of a given number of aircraft is called a run. The MOE (Measue Of Effectiveness) for a run of REGAME, the Probability of Survival, is the number of replications which yield 100 percent aircraft killed divided by the number of replications in that run. The number of replications required to achieve sufficient confidence in the MOE can be established only after some experience has been gained with applying the model to similar problems. The mean, variance, and standard deviation of

the number of missiles fired, and the number of kills is provided in the output of the program.

III. CHANGES TO REGAME

A. MOTIVATION

The simulation desired was one which would be realistic for present day applications. It had to be a relatively small scale model to allow for the use of readily available input data, and yield a result which had a general usage. It had to be one which was applicable to a general type of ship, such as destroyers, rather than one which was applicable to a specific class of ship, such as DDG-2 destroyers. The simulation had to have the characteristic of relatively short run time so that rough solutions to a scenario could be evaluated without expending sizeable amounts of computer time. It had to relate to a navigational plot and play more than three ships and twenty missiles. Air-to-air combat had to be played since it was an integral part of carrier air defense. In other words, SEAGUARD had to accommodate the characteristics of ASMD and maintain the desired general approach to the simulation.

B. CHANGE IN THE PLAYING AREA

The evaluation of ASMD of a carrier task force required the capability of evaluating the defense of a sector as small as a few degrees or as great as a full circle. To facilitate the use of a full circle and the preliminary calculations required to set up the disposition of defensive

forces, the playing area was reoriented to a navigational plot where north, 000 degrees, is represented by the positive y axis in the x,y plane. This change allows the user to calculate his input positions relative to the center of the ship formation and locate the center of the playing area at any point in the x,y plane. The only change in the computer code necessitated by this playing area change was the method of computing the entry points of the offensive weapons into the playing area. REGAME calculated these points by generating an angle θ relative to the bisector of the central playing angle using the Monte Carlo technique. The coordinates of the entry points (X,Y) were calculated as follows:

Let

$$A = RCC \times \cos \theta$$

$$B = RCC \times \sin \theta$$

Then for positive values of θ

$$X = GX + A$$

$$Y = GY + B$$

For negative values of θ

$$X = GX + A$$

$$Y = GY - B$$

where RCC, GX, and GY are defined in Figure 1.

The central playing angle of SEAGUARD is bisected by the y axis, 000 degrees. To generate the entry points for the ASMs, an angle ϕ is generated using the Monte Carlo technique. The lower bound on ϕ is 360 degrees minus one half the value of the central playing angle, and the upper bound on ϕ is 360 degrees plus one half the value of the central playing angle. The coordinates of the entry points are calculated as follows:

Let

$$A = RCC \times \cos(\tfrac{1}{2}\pi - \phi)$$

$$B = RCC \times \sin(\tfrac{1}{2}\pi - \phi)$$

Then

$$X = GX + A$$

$$Y = GY + B$$

The transformation $(\frac{1}{2}\pi - \phi)$ accomplished the change from the cartesian coordinate system to the navigational plot.

This change enables the user to take advantage of the full x,y plane, facilitates orientation of the ship formation to (0,0), and allows an incoming attack to approach from any direction.

C. INTRODUCTION OF AIR PLATFORMS

The play of air platforms as CAP (Combat Air Patrol) was essential in evaluating ASMD of a carrier task force. The addition of air platforms was accomplished by adding altitude to the input parameters for the defensive missile platforms. This change in addition to increasing the size of arrays slightly, required changes in computing the radar horizon. Slant range, the vector sum of the range and relative altitude, had to be used in the fire event. The use of CAP also required the development of a method to model the real effect, of passing detection information, on the ASM-surface platform encounter. The factors which are important to the outcome of the ASM-surface platform encounter are tracking radar range, missile range, and radar horizon. If the track of the ASM were known to the surface platform, as would be the case if the CAP informed him, the only limiting factors of importance in the ASM-surface platform encounter would be the tracking radar range and the range of the defensive missile. In SEAGUARD, to achieve this effect, the longest radar horizon of any of the defensive platforms is used as the radar horizon for all defensive platforms; therefore, the play of the ASM-surface platform encounter resembles the action of passing detection information among the defensive

units. If a defensive formation is used such that the defensive platforms at the outer range of the force are not positioned symmetrically, in some cases the lifting of the firing restriction due to the radar horizon would yield a result not representative of the real world situation. Therefore it is assumed that the disposition of forces is symmetrical to the threat axis and that detection information is passed among all defensive units.

D. RELOAD EVENT

Due to the addition of air platforms into the simulation, the reload event had to be altered to accommodate for the unique character of an air platform reload. The sequence that SEAGUARD follows to reload air platforms is as follows:

(1) Determine if the missile site is an air platform.

(2) If the site is an air platform, determine if all its missiles have been fired.

(3) If all its missiles have been fired, schedule CAP relief.

This sequence is based on the assumption that an aircraft will not reload until it has fired all its missiles. The reload event then schedules the relief of the on station aircraft with a replacement aircraft. The input parameters for reload time, in the case of defensive missiles sites which are aircraft, should represent the time required to relieve an on station aircraft. The reload event for the defensive ground site is accomplished in the same manner as REGAME.

E. OFFENSIVE AND DEFENSIVE WEAPONS

The number of offensive weapons was increased from 20 to 50. This increase was accomplished by increasing the array size for those arrays associated with offensive weapons. This rendered the model capable of analyzing a larger offensive threat. The use of ASMs in place of aircraft required no change to the code of REGAME since the input limits for the aircraft were already great enough to accommodate the characteristics of missiles and since the altitude profile of the ASM was assumed to be flat until the ASM was within close range of its target.

The number of defensive missile sites was increased from 3 to 10. This increase was also accomplished by increasing the size of the respective arrays. Associated with this increase in the defensive missile sites and the change in playing area, the firing event was modified to include a check against firing a defensive missile over the position of the carrier or capital ships in the center of the playing area. If the relative difference between the bearing to the center of the playing area, is less than 45 degrees, the missile site will not fire at the incoming ASM. This check prevents the occurrence of a defensive missile hit on the carrier or capital ships. The firing event was also modified to account for the difference in altitude between the ASM and the SAM or AAM. The CAP also has the capability of shooting up or down at the ASM.

F. INPUT/OUTPUT

The input to SEAGUARD is arranged so that the defensive missile site parameters are inputted on the first ten cards, one missile area per card. The ASM input and the playing area input follow on the next two input cards respectively.

On output, the MOE, called the Probability of Survival, for REGAME was deemed inappropriate for SEAGUARD because it is based on the assumption that one offensive weapon which penetrated the defense rendered that replication a failure for the defense. An MOE which would provide a more refined measure of marginal product was desired. SEAGUARD in addition to providing the Probability of Survival, now called the Probability of Zero Hits, provides an MOE called Percent Killed, which is the number of kills of offensive missiles divided by the total number of offensive missiles played. This new measure provides more sensitivity to the outcome over a broader spectrum of threat.

F. INPUT/OUTPUT

The input to SEAGUARD is arranged so that the defensive missile site parameters are inputted on the first ten cards, one missile area per card. The ASM input and the playing area input follow on the next two input cards respectively.

On output, the MOE, called the Probability of Survival, for REGAME was deemed inappropriate for SEAGUARD because it is based on the assumption that one offensive weapon which penetrated the defense rendered that replication a failure for the defense. An MOE which would provide a more refined measure of marginal product was desired. SEAGUARD in addition to providing the Probability of Survival, now called the Probability of Zero Hits, provides an MOE called Percent Killed, which is the number of kills of offensive missiles divided by the total number of offensive missiles played. This new measure provides more sensitivity to the outcome over a broader spectrum of threat.

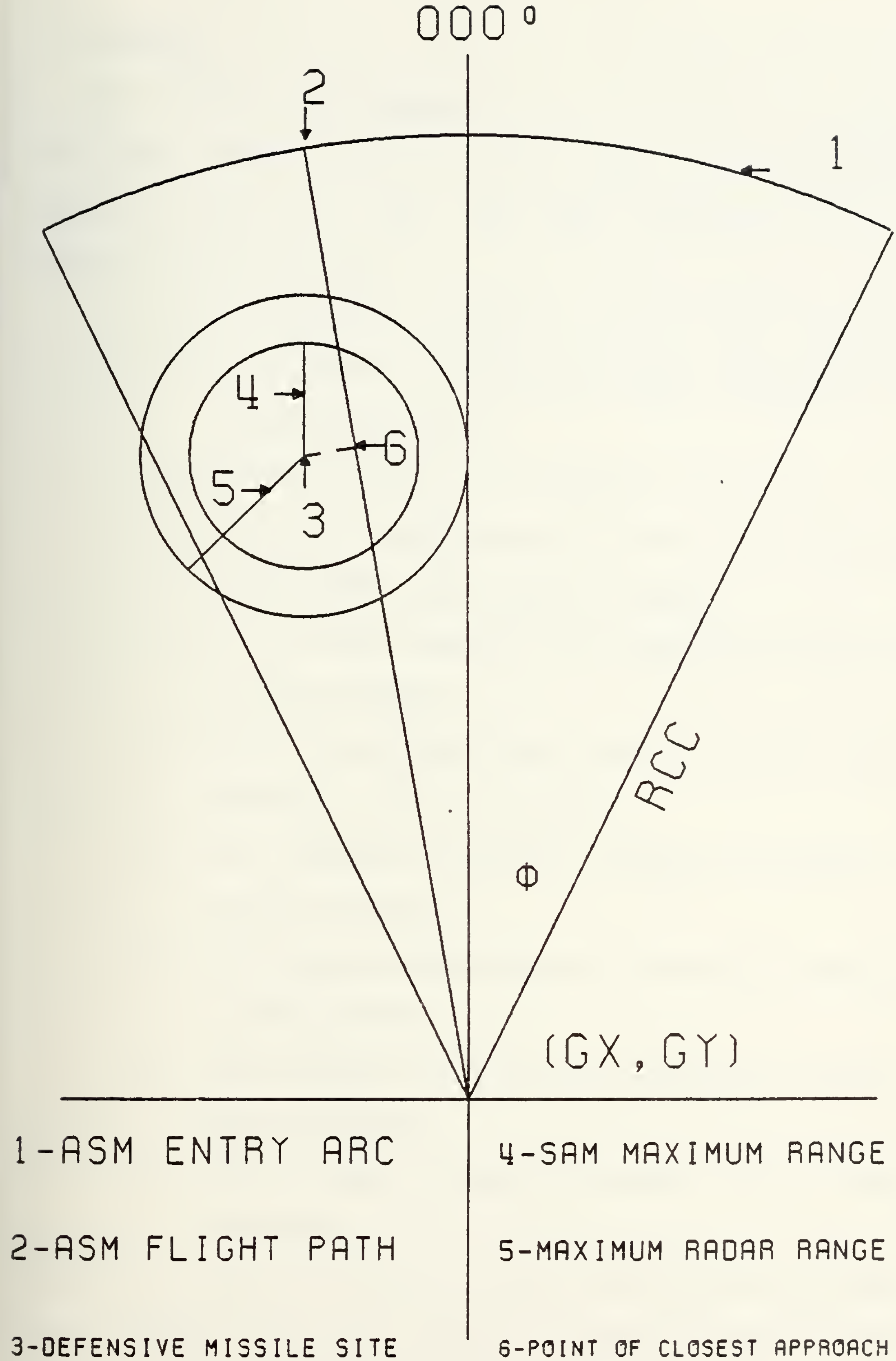
IV. SEAGUARD

A. PLAYING AREA

The playing area for SEAGUARD as illustrated in Figure 3 is a circle or portion of a circle whose center, radius, and central playing angle are inputs. The radius, RCC, can be as large as 1000 miles. The arc length of the playing area can vary from 0 to 360 degrees and is also an input parameter. The bisector of the central playing angle must be in the direction of the positive y axis corresponding to 000 degrees or north. Defensive missile sites may be placed outside the playing area; however, only actions which occur inside the playing area affect the outcome of the simulation.

B. OFFENSE

The offense consists of as many as 50 ASMs which are directed toward the vertex of the central playing angle. The missiles originate uniformly along the outer arc of the playing area. ASM speed, ASM altitude, and the separation time between ASMs are uniformly distributed between bounds specified by the user on input. The ASMs play a passive role in the simulation in that they serve only as a stimuli for the ASMD simulation. An ASM is assumed to have hit the carrier or capital ships, if it reaches the central point of the playing area.



SEAGUARD PLAYING AREA

Figure 3

C. DEFENSE

The defense for SEAGUARD is a set of up to 10 missile platforms, either surface or air, which can be positioned anywhere in the x,y plane. The input parameters for the defense are as follows:

(1) The number of missile areas in the simulation.

(2) The number of tracking radars assigned to each missile platform.

(3) The altitude of the missile platform.

(4) The grid coordinates of each missile platform.

(5) The search radar maximum range.

(6) The average speed of the defensive missile for each platform.

(7) The single salvo kill probability for the missile on each platform.

(8) The number of missile launchers assigned to each platform.

(9) The minimum and maximum acquisition time for each platform.

(10) The minimum and maximum assessment time for each platform.

(11) The minimum and maximum reload time for each platform.

Acquisition time is defined as the time required to pass an active target from search radar to tracking radar. Assessment time is defined as the time a fire control radar must remain locked on the target to assess the result of the

attempted interception. The defensive sites detect and fire at ASMs based on the missile area characteristics and the schedule of events. A fire event will not be scheduled for the defense if the point of closest approach of the ASM is within 45 degrees of bearing to the capitol ships or carrier in the center of the playing area. Either coordinated or uncoordinated fire procedures are available for the play of SEAGUARD.

D. GAME DOCTRINE

SEAGUARD simulates the interactions between offensive ASMs and defensive SAMs or defensive AAMs. This task is accomplished by an event step simulation. A series of events and associated times are generated and listed in a chronologically ordered schedule. There are five types of events which are as follows:

(1) Fire missile salvo.

(2) Missile intercept.

(3) Reload missile launcher.

(4) Free tracking radar.

(5) Change of engagement status.

The simulation initially generates the ASM entry points into the playing area. The ASMs proceed directly to the center of the playing area. The SAM and AAM missile firing events are then generated and stored for the missile sites which are encountered. After this initial generation of events, the simulation begins by executing the schedule of events

(4) Time spacing between ASMs.

(5) ASM speed.

(6) ASM altitude.

(7) Entry points of ASMs.

(8) Determination of ASM kill.

In short, SEAGUARD is a series of computer calculations which simulate ASMD through the generation of events and the modification of those events through checks of capability and availability.

E. INPUT

SEAGUARD is structured so that the input data required to run the simulation must be inputed using 12 computer cards whose format is described in Tables I and II. Table I lists the input format for the missile sites which are inputed as cards 1 through 10. If a missile site is not used, a blank card must be inserted into the data deck. Table II lists the input format for the ASM inputs and the playing area inputs respectively. Standard FORTRAN format is used; therefore, each integer value must be right justified within its allotted field. All real inputs are floating point. If the decimal is not included somewhere in the field, the real number must be right justified also. Each set of 12 cards defines a game. A game can consist of a run of a fixed number of ASMs for the desired number of replications or it can consist of a series of runs with the number of ASMs being increased until the maximum number of ASMs is

TABLE I

MISSILE SITE INPUT FORMAT

COLUMNS	NAME	LOWER BOUND	UPPER BOUND	UNITS
1	MTRF(I)	0	100	
2	MAL(I)	0	10	
3-8	AX(I)	-1000	1000	MILES
9-14	AY(I)	-1000	1000	MILES
15-17	RMR(I)	0	1000	MILES
18-20	RMAX(I)	0	1000	MILES
21-25	AVS(I)	0	10000	MILES/HR.
26-29	PX(I)	0	100	PERCENT
30-33	ATM(I)	0	1000	MINUTES
34-37	ATX(I)	0	1000	MINUTES
38-41	ASM(I)	0	1000	MINUTES
42-45	ASX(I)	0	1000	MINUTES
46-49	RTM(I)	0	1000	MINUTES
50-53	RTX(I)	0	1000	MINUTES
54-57	CALT(I)	0	100	K FEET

TABLE II

ASM AND PLAYING AREA INPUT FORMAT

COLUMNS	NAME	LOWER BOUND	UPPER BOUND	UNITS
1-5	BSX	0	10000	MILES/HR.
6-10	BSM	0	10000	MILES/HR.
11-14	BTX	0	1000	MINUTES
15-18	BTM	0	1000	MINUTES

COLUMNS	NAME	LOWER BOUND	UPPER BOUND	UNITS
1-2	NMISAR	0	10	
3	IEND	0,1		
4-5	NBMAX	0	50	
6-7	NBOMB	0	50	
8-9	NBDEL	0	50	
10-15	GX	-1000	1000	MILES
16-21	GY	-1000	1000	MILES
22-27	RCC	0	1000	MILES
28-32	CPA	0	360	DEGREES
33-34	TMAX	0	100	HOURS
35-36	NRPL	0	20	
37-40	AMAX	0	100	K FEET
41-44	AMIN	0	100	K FEET
45-48	STEP	0	6000	MINUTES
49	NFLAG	0,1,2		
50	IHIST	0,1		
51	OUTPUT	0,1		

played or until the user inputed upper limit on the number of ASMs is played. To vary any other parameter of the game, a new set of 12 cards or a series of 12 card sets must be added to the data deck. The definition of the input variables for SEAGUARD are listed in Table III.

F. OUTPUT

SEAGUARD yields three products as output, namely, the Battle History, the Standard output, and the Summary output. These products may or may not be produced according to the desires of the user with the exception that the Summary output will always be produced. The Battle History is a listing of events. This list is produced from two program elements, one called SNE, meaning schedule next event, and the other called TNE, meaning take next event. Table IV depicts a typical Battle History. The format for the events listed in the Battle History is as follows:

TIME OF EVENT	TYPE OF EVENT	ASM #	MISSILE AREA #

The time appears in hours and the type event, ASM, and missile area are identified by number. The code for the type of event is as follows:

NUMBER	EVENT
1	Reload
2	Set tracking radar free
3	Intercept
4	Change engagement status
5	Fire

TABLE III

DESCRIPTION OF INPUTS

VARIABLE	DESCRIPTION
MTRF(I)	The number of tracking radars assigned to missile area I.
MAL(I)	The number of missile launchers assigned to missile area I.
AX(I)	The x-coordinate of missile area I.
AY(I)	The y-coordinate of missile area I.
RMR(I)	The search radar maximum range of missile area I.
RMAX(I)	The missile maximum range for missile area I.
AVS(I)	The average speed of the SAM for missile area I.
PK(I)	The one shot probability of kill for the SAM or AAM of missile area I.
ATM(I)	The minimum acquisition time for missile area I.
ATX(I)	The maximum acquisition time for missile area I.
ASM(I)	The minimum assessment time for missile area I.
ASX(I)	The maximum assessment time for missile area I.
RTM(I)	The minimum reload time for missile area I.
RTX(I)	The maximum reload time for missile area I.
CALT(I)	The platform altitude for missile area I.
BSX	The maximum ASM speed.
BSM	The minimum ASM speed.
BTX	The maximum time spacing between ASMs.
BTM	The minimum time spacing between ASMs.
NMISAR	The number of missile areas in the simulation.

VARIABLE	DESCRIPTION
IEND	To change the value of any input parameter for another run of SEAGUARD, another set of 12 input cards is required. The new set follows the original set in the data deck. IEND is a program flag used to indicate the last set of data. 0 Last set of data. 1 Another set of data follows.
NBMAX	The maximum number of ASMs in the last run.
NBOMB	The number of ASMs in the first run.
NBDEL	The increment of the increase of ASMs between runs. The number of runs to be made with one set of input cards will be the minimum integer value of N such that NBOMB + (N) x (NBDEL) = NBMAX
GX	The x-coordinate of the center.
GY	The y-coordinate of the center.
RCC	The length of the radius of the playing area.
CPA	The central playing angle.
TMAX	The maximum game time.
NRPL	The number of replications to be played.
AMAX	The maximum ASM altitude.
AMIN	The minimum ASM altitude.
STEP	STEP is a time value used for computer efficiency. When a fire event cannot be executed, a new fire event is scheduled at the current time plus the value of STEP.
NFLAG	This value designates the missile firing procedure as follows:

TABLE III - CONTINUED

VARIABLE	DESCRIPTION
	0 Both coordinated and uncoordinated procedures are used in the simulation.
	1 Uncoordinated procedure only.
	2 Coordinated procedure only.
IHIST	This value designates whether the Battle History is desired for output as follows:
	0 No Battle History
	1 Battle History output is printed.
OUTPUT	This value indicates whether the summary output or both the standard and the summary output is desired as follows:
	0 Standard and summary output.
	1 Summary output only.

TABLE IV

BATTLE HISTORY

		1.103	5	1	3
		1.163	5	1	7
		1.150	5	1	9
		1.194	5	1	10
		1.095	5	2	3
		1.135	5	2	7
		1.120	5	2	9
		1.157	5	2	10
		1.096	5	3	3
		1.139	5	3	7
		1.188	5	3	10
		1.076	5	4	1
		1.132	5	4	5
		1.145	5	4	7
		1.165	5	4	10
		1.121	5	5	2
		1.102	5	5	3
		1.143	5	5	7
		1.127	5	5	9
		1.163	5	5	10
1.076	5	4	1		
		1.099	3	4	1
1.095	5	2	3		
		1.119	3	2	3
1.096	5	3	3		
		1.120	3	3	3
1.099	3	4	1		
		1.106	5	4	1
		1.106	2	4	1
		1.106	4	4	1
1.102	5	5	3		
		1.128	3	5	3
1.103	5	1	3		
		1.129	3	1	3
1.106	2	4	1		
1.106	4	4	1		
1.106	5	4	1		
		1.116	3	4	1
1.116	3	4	1		

The events are ordered such that if two events occur at the same time, the event with the lowest event code number would be executed first. The indented lines represent events that are being scheduled by the SNE and the lines nearest the margin are events that are being executed by the TNE. The Battle History presented in Table IV is read in the following manner. The initial entries of the Battle History are the fire events generated by the SNE. The first event to be executed by the TNE is at time 1.076. This is a fire event associated with ASM number 4 and missile are number 1. Execution of this fire event results in the scheduling of an intercept event at time 1.099 as is illustrated in the following line of the Battle History which is indented. At time 1.099, the intercept event is executed by TNE and results in the scheduling of a new fire event, a free tracking radar event, and a change engagement status event, indicating that the defensive missile fired at time 1.076 was not a kill. The Battle History can be an effective tool in determining if the simulation is functioning in the manner that was desired. The program run time is greatly increased when the Battle History is desired as output. Table V is an explanation of the information detailed on the Standard output. Table VI illustrates the format of the Standard output. Computer time is increased when the Standard output is desired.

The Summary output is illustrated in Tables VII and VIII. It is a summary of the actions in each replication of the run. It lists the statistics for each replication and for

TABLE V

DEFINITION OF OUTPUT HEADINGS

HEADING	DEFINITION
ASM	The ASM number.
SMAX	The total number of salvos fired at this ASM by missile area x, where x varies from 1 to 10.
TOTAL	The total number of salvos fired at this ASM by all missile area.
A/K	The number of the missile area killing the ASM. A/K = 0 indicates that the ASM penetrated the defense.
XCOORD	The x-coordinate of the ASM entry point in miles.
YCOORD	The y-coordinate of the ASM entry point in miles.
ASMH	The radar horizon in miles for the ASM.
ALTITUDE	The ASM altitude in thousands of feet.
SPEED	The ASM speed in miles per hour.
TIME	The ASM entry time in hours.
TIME(X)	The earliest possible time, in hours, that missile area x can fire at the ASM. Where x varies from 1 to 10. A zero value indicates that the missile area cannot fire at this ASM.

TABLE VI

STANDARD OUTPUT

COORDINATED ATTACK RESULTS   REP 15

ASM	XCOORD	YCOORD	ASMH	MALT	ASM SPEED
1	-213.65	-338.16	144.55	2.37	2250.27
2	-14.95	-399.72	139.62	1.48	2016.38
3	261.84	302.39	146.28	2.69	2347.22
4	-41.65	357.83	138.81	1.33	2096.11
5	-16.73	-399.65	136.86	0.99	2266.78

ASM	TIME	TIME(1)	TIME(2)	TIME(3)	TIME(4)	TIME(5)	TIME(6)	TIME(7)	TIME(8)
1	1.000	0.0	1.093	1.150	0.0	0.0	1.142	1.161	1.132
2	1.000	0.0	1.127	1.132	0.0	0.0	1.171	1.171	1.160
3	1.000	1.094	0.0	1.133	0.0	1.127	1.159	1.128	0.0
4	1.000	1.086	1.170	1.173	1.150	0.0	1.167	1.171	0.0
5	1.000	0.0	1.115	1.119	0.0	0.0	1.153	1.154	1.141

ASM	SMA1	SMA2	SMA3	SMA4	SMA5	SMA6	SMA7	SMA8	SMA9	SMA10	TOTAL	A/K
1	0	2	0	0	0	0	0	0	0	0	2	2
2	0	2	0	0	0	0	0	0	0	0	2	2
3	2	0	0	0	0	0	0	0	0	1	3	0
4	2	0	0	0	0	0	0	0	0	0	2	1
5	0	2	0	0	0	0	0	0	0	0	2	2

TABLE VII

SUMMARY OUTPUT PART 1

SUMMARY COORDINATED RESULTS

ASM S FIRED

ASM S	REP	MA1	MA2	MA3	MA4	MA5	MA6	MA7	MA8	MA9	MA10	TOTAL	MEAN	VAR	SD
5	1	4.	0.	6.	0.	1.	0.	0.	0.	0.	1.	12.00	1.20	3.96	1.99
5	2	6.	0.	3.	2.	1.	0.	0.	0.	0.	1.	13.00	1.30	3.41	1.85
5	3	5.	6.	0.	0.	1.	0.	1.	0.	0.	1.	14.00	1.40	4.44	2.11
5	4	5.	1.	4.	2.	0.	0.	1.	0.	0.	1.	14.00	1.40	2.84	1.69
5	5	4.	2.	3.	0.	0.	0.	1.	0.	0.	2.	12.00	1.20	1.96	1.40
5	6	4.	3.	5.	0.	0.	1.	2.	0.	0.	1.	16.00	1.60	3.04	1.74
5	7	5.	1.	4.	0.	1.	0.	1.	0.	0.	1.	13.00	1.30	2.81	1.68
5	8	2.	3.	6.	2.	0.	2.	0.	0.	0.	0.	15.00	1.50	3.45	1.86
5	9	6.	3.	2.	0.	0.	1.	0.	0.	0.	1.	13.00	1.30	3.41	1.85
5	10	5.	0.	4.	0.	0.	0.	0.	0.	0.	2.	11.00	1.10	3.29	1.81
5	11	2.	3.	1.	0.	1.	0.	1.	0.	0.	0.	8.00	0.80	0.96	0.98
5	12	3.	5.	1.	0.	0.	1.	0.	0.	0.	1.	11.00	1.10	2.49	1.58
5	13	0.	6.	5.	0.	0.	2.	1.	0.	0.	1.	15.00	1.50	4.45	2.11
5	14	2.	2.	5.	0.	0.	0.	3.	0.	0.	2.	14.00	1.40	2.64	1.62
5	15	4.	6.	0.	0.	0.	0.	0.	0.	0.	1.	11.00	1.10	4.09	2.02

	MA1	MA2	MA3	MA4	MA5	MA6	MA7	MA8	MA9	MA10	TOTAL
MEAN	3.80	2.73	3.27	0.40	0.33	0.47	0.73	0.0	0.0	1.07	12.80
VAR	2.69	4.46	3.93	0.64	0.22	0.52	0.73	0.0	0.0	0.33	3.89
SD	1.64	2.11	1.98	0.80	0.47	0.72	0.85	0.0	0.0	0.57	1.97

TABLE VIII

SUMMARY OUTPUT PART 2

ASM KILLS

ASM S	REP	MA1	MA2	MA3	MA4	MA5	MA6	MA7	MA8	MA9	MA10	TOTAL	MEAN	VAR	SD
5	1	0.	0.	3.	0.	0.	0.	0.	0.	0.	1.	4.00	0.40	0.84	0.92
5	2	1.	0.	2.	0.	1.	0.	0.	0.	0.	1.	5.00	0.50	0.45	0.67
5	3	1.	2.	0.	0.	0.	0.	0.	0.	0.	0.	3.00	0.30	0.41	0.64
5	4	0.	0.	1.	0.	0.	0.	0.	0.	0.	0.	1.00	0.10	0.09	0.30
5	5	0.	1.	0.	0.	0.	0.	1.	0.	0.	1.	3.00	0.30	0.21	0.46
5	6	2.	0.	0.	0.	0.	1.	0.	0.	0.	0.	3.00	0.30	0.41	0.64
5	7	0.	1.	2.	0.	0.	0.	0.	0.	0.	0.	3.00	0.30	0.41	0.64
5	8	0.	0.	3.	1.	0.	1.	0.	0.	0.	0.	5.00	0.50	0.85	0.92
5	9	1.	1.	0.	0.	0.	0.	0.	0.	0.	1.	3.00	0.30	0.21	0.46
5	10	2.	0.	1.	0.	0.	0.	0.	0.	0.	0.	3.00	0.30	0.41	0.64
5	11	0.	3.	1.	0.	0.	0.	0.	0.	0.	0.	4.00	0.40	0.84	0.92
5	12	3.	0.	0.	0.	0.	0.	0.	0.	0.	1.	4.00	0.40	0.84	0.92
5	13	0.	2.	0.	0.	0.	0.	1.	0.	0.	0.	3.00	0.30	0.41	0.64
5	14	1.	0.	1.	0.	0.	0.	0.	0.	0.	1.	3.00	0.30	0.21	0.46
5	15	1.	3.	0.	0.	0.	0.	0.	0.	0.	0.	4.00	0.40	0.84	0.92

	MA1	MA2	MA3	MA4	MA5	MA6	MA7	MA8	MA9	MA10	TOTAL
MEAN	0.80	0.87	0.93	0.07	0.07	0.13	0.13	0.0	0.0	0.40	3.40
VAR	0.93	1.18	1.13	0.06	0.06	0.12	0.12	0.0	0.0	0.24	0.91
SD	0.91	1.09	1.06	0.25	0.25	0.34	0.34	0.0	0.0	0.49	0.95

PROBABILITY OF ZERO HITS = 0.13

PERCENT ASM'S KILLED = 0.63

each missile area. At the end of the Summary output, the Probability of Zero Hits and the Percent Killed are listed.

V. OUTPUT ANALYSIS

A. STANDARD SET UP

The output analysis of a simulation model must start with the specification of initial conditions. This set of initial conditions or what is called the standard set up was decided upon after some experience was gained with the model in the evaluation of a few general scenarios. The standard set up is not meant to model any particular real world situation. It is a general composite of various scenarios derived in a manner such that the usual intervals of interest for the input parameters would be represented in the output analysis. The standard set up is illustrated in Table IX. The definition of each variable in this table was given in Table III.

B. PARAMETRIC ANALYSIS

The parametric analysis studies the measure of effectiveness as a function of the input parameters. Figure 4 shows the MOE as a function of p(k), probability of kill. Each of the lines on the graph represents the particular raid size as denoted by the symbols listed. The graph shows the expected result that the MOE increases with an increase in the single shot probability of kill of the defensive missiles. It is interesting to note that the result for the 30, 40, and 50 ASM threat sizes are not significantly different, the result

TABLE IX

STANDARD INPUT

GAME PARAMETERS

XCOORDINATE OF CENTER	(GX) =	0.0	MILES
YCOORDINATE OF CENTER	(GY) =	0.0	MILES
RADIUS OF CIRCLE	(RCC) =	400.00	MILES
CENTRAL PLAYING ANGLE	(CPA) =	90.00	DEGREES

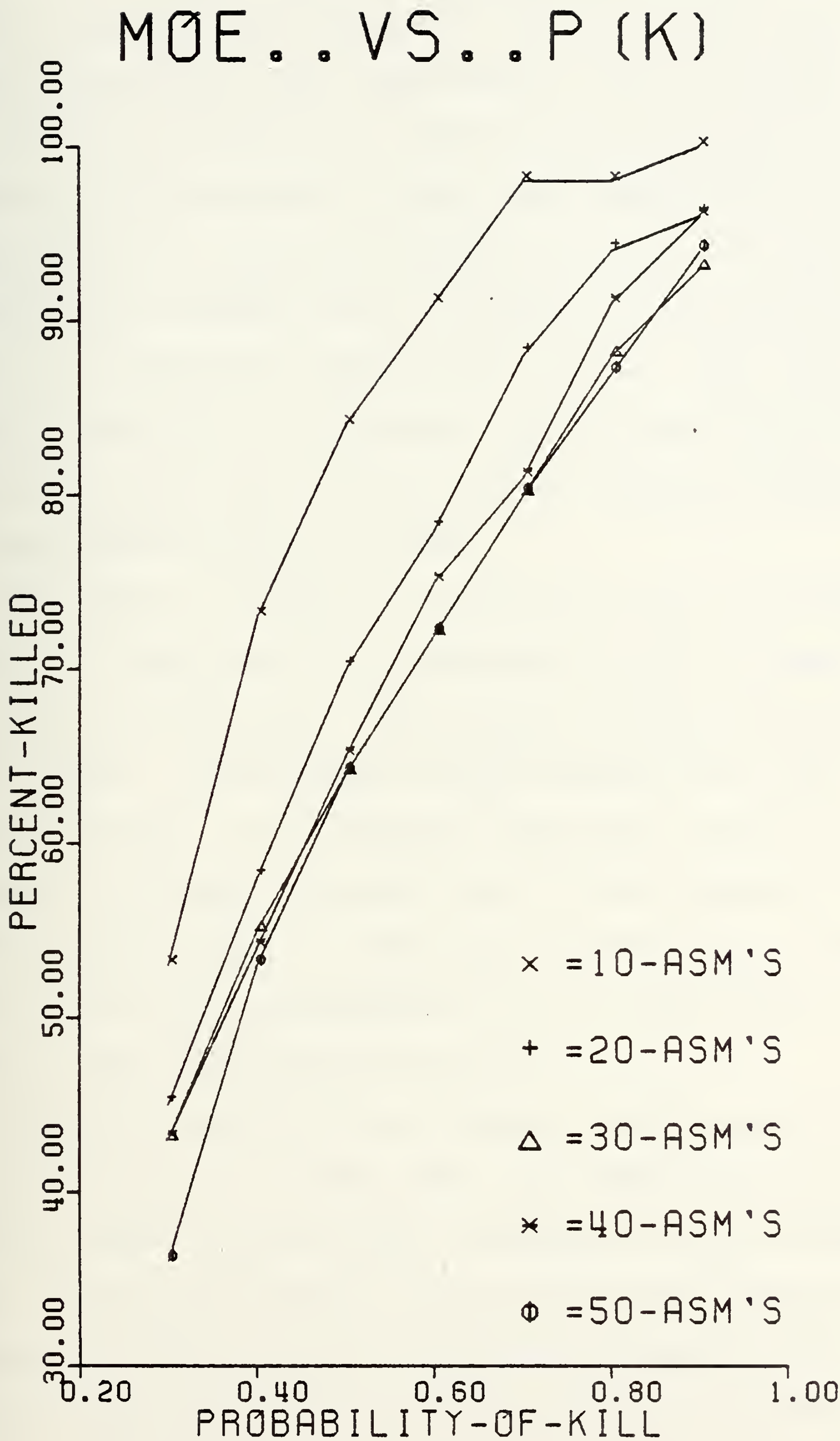
MISSILE AREA INFORMATION

MISSILE AREA NR

1	2	3	4	5	6	7	8	9	10		
0.0	-85.00	85.00	-25.00	25.00	-50.00	50.00	-25.00	25.00	0.0	MILES	AX
100.00	-50.00	-50.00	42.50	42.50	0.0	0.0	-42.50	-42.50	0.0	MILES	AY
11.00	11.00	11.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	K FT	CALT
6	6	6	2	2	3	3	2	2	1		MAL
6	6	6	2	2	3	3	2	2	1		MTRF
125.00	125.00	125.00	80.00	80.00	80.00	80.00	100.00	100.00	100.00	MILES	RMR
125.00	125.00	125.00	25.00	25.00	65.00	65.00	25.00	25.00	5.00	MILES	RMAX
3000.00	3000.00	3000.00	2000.00	2000.00	2000.00	2000.00	2000.00	2000.00	2000.00	MI/HR	AVS
0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	MINS	ATM
0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	MINS	ATX
0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	MINS	ASM
0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	MINS	ASX
5.00	5.00	5.00	0.10	0.10	0.10	0.10	0.10	0.10	0.10	MINS	BTM
90.00	90.00	90.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	MINS	BTX
60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	PER.	PK

ASM INFORMATION

MAXIMUM SPEED (BSX)=2500.0 MILES/HR
MINIMUM SPEED (BSM)=2000.0 MILES/HR
MAXIMUM ALTITUDE (AMAX)= 2.0 K FEET
MINIMUM ALTITUDE (AMIN)= 0.0 K FEET
MAXIMUM SEPERATION TIME (BTX)= 0.85 MINS.
MINIMUM SEPERATION TIME (BTM)= 0.65 MINS.



MØE..VS..P(K)
PERCENT-KILLED
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
0.20
0.40
0.60
0.80
1.00
PROBABILITY-OF-KILL
× =10-ASM'S
+ =20-ASM'S
Δ =30-ASM'S
⋇ =40-ASM'S
Φ =50-ASM'S

Figure 4

for the 20 ASM threat shows a slightly greater displacement from the x-axis, and the result for the 10 ASM threat size shows considerably more displacement from the x-axis.

Figure 5, the MOE as a function of threat size illustrates this difference a little more dramatically. In Figure 5, this difference appears as a greater slope in the curves at the threat sizes 10 through 30 and a flattening out of the curve for larger threat sizes. Each curve represents a given p(k) as labelled. These results show that the saturation raid size for this general scenario appears to be approximately 25 to 30 ASMs.

The MOE as a function of reload time is illustrated by Figure 6. Again each line represents a given threat size as annotated on the graph. This format is repeated throughout the chapter. As expected the MOE decreases with the increase in reload time. An interesting additional result is that the effect of a change in reload time is more dramatic as the threat size increases. The change in MOE does not appear significant until approximately 20 to 30 ASM threat size, showing again saturation raid size.

MOE as a function of the central playing angle is illustrated in Figure 7. This result may appear surprising because as the central playing angle increases the MOE increases. This result is believable realizing that the threat size remains constant, and now the offensive missiles will probably be tracked and fired upon by additional defensive sites.

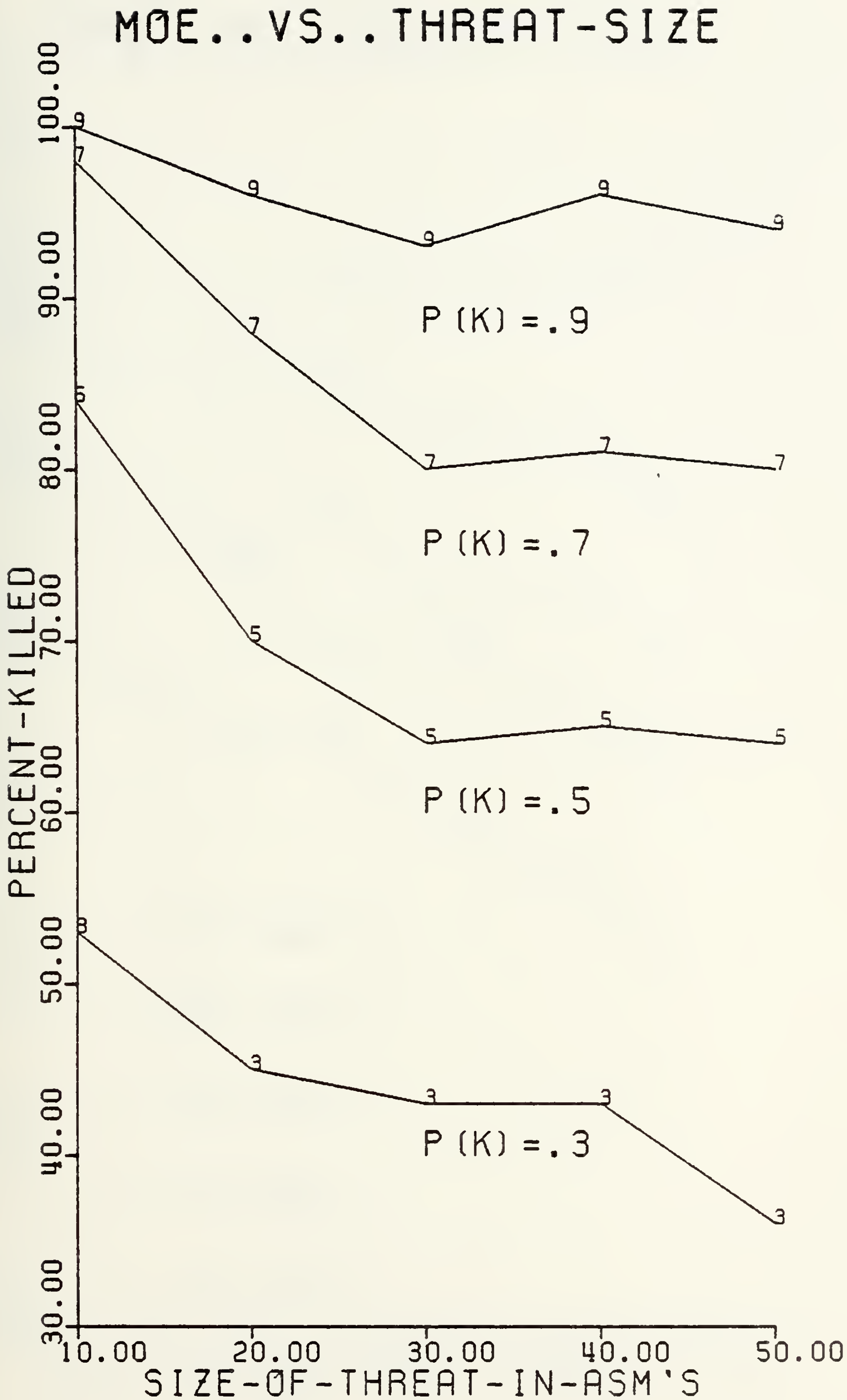

MØE..VS..THREAT-SIZE
PERCENT-KILLED
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
10.00
20.00
30.00
40.00
50.00
SIZE-ØF-THREAT-IN-ASM'S
P(K)=.9
P(K)=.7
P(K)=.5
P(K)=.3

Figure 5

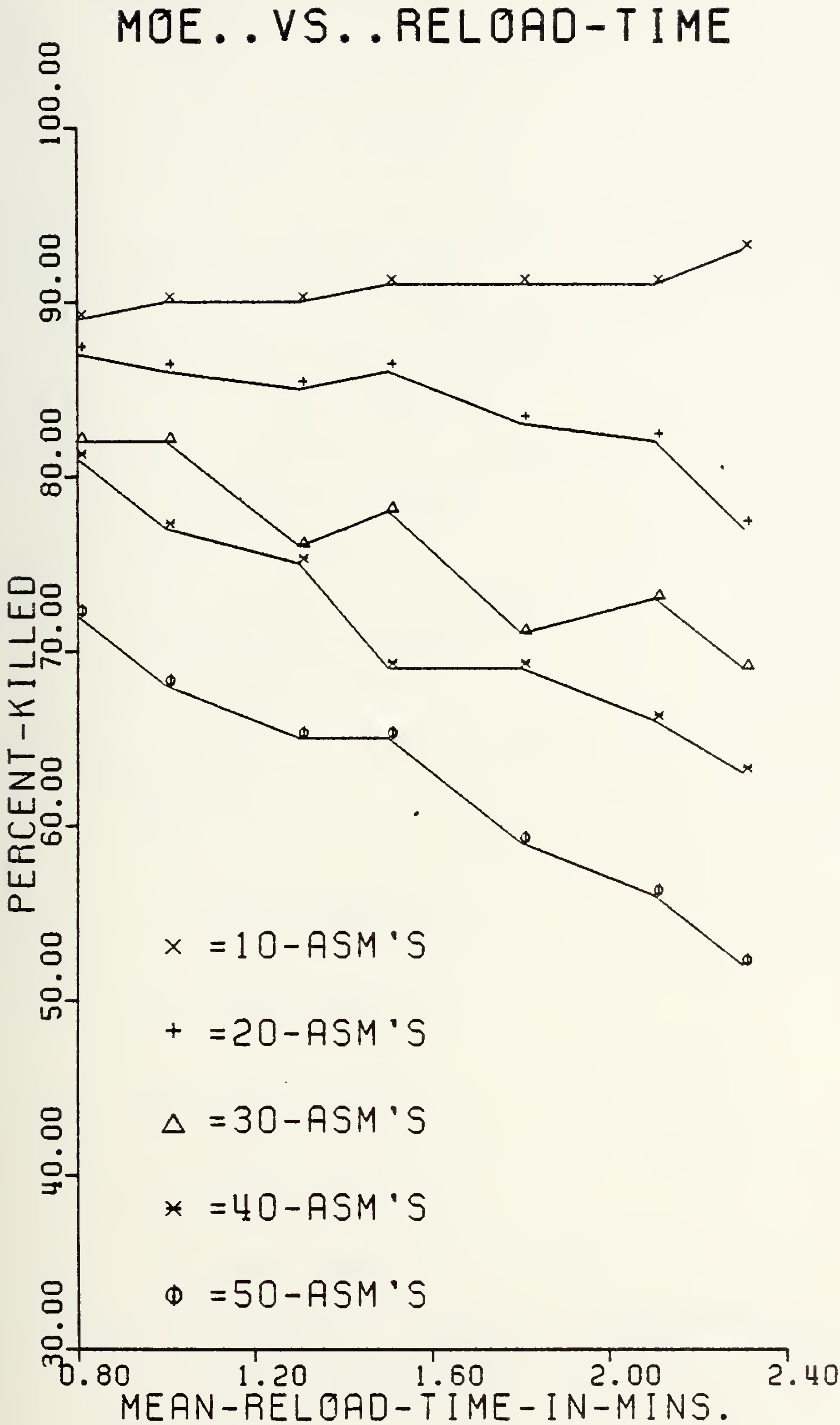

MOE..VS..RELOAD-TIME
PERCENT-KILLED
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
0.80
1.20
1.60
2.00
2.40
MEAN-RELOAD-TIME-IN-MINS.
× =10-ASM'S
+ =20-ASM'S
△ =30-ASM'S
✳ =40-ASM'S
Φ =50-ASM'S

Figure 6

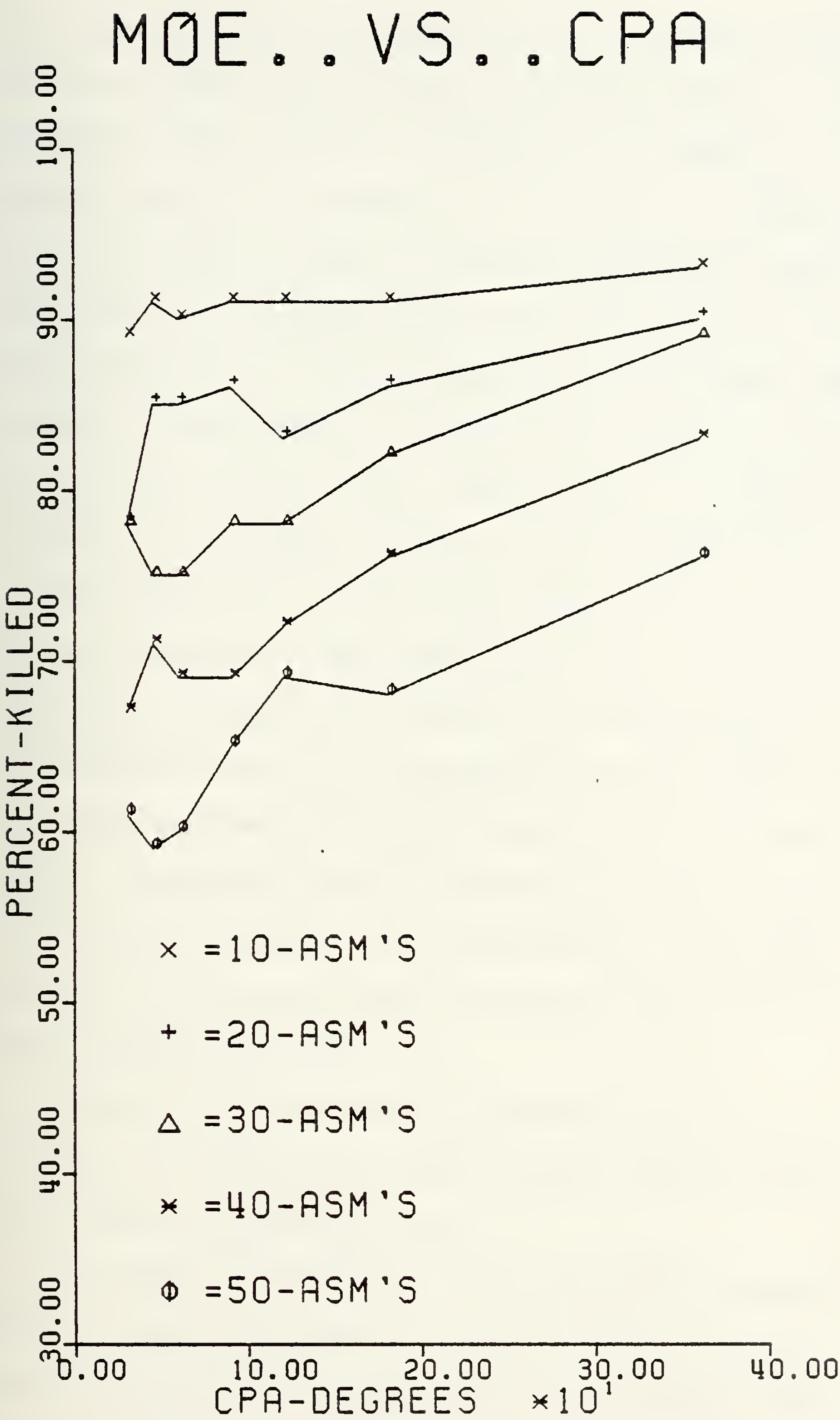

MØE..VS..CPA
PERCENT-KILLED
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
0.00
10.00
20.00
30.00
40.00
CPA-DEGREES ×10^1
× =10-ASM'S
+ =20-ASM'S
△ =30-ASM'S
✱ =40-ASM'S
Φ =50-ASM'S

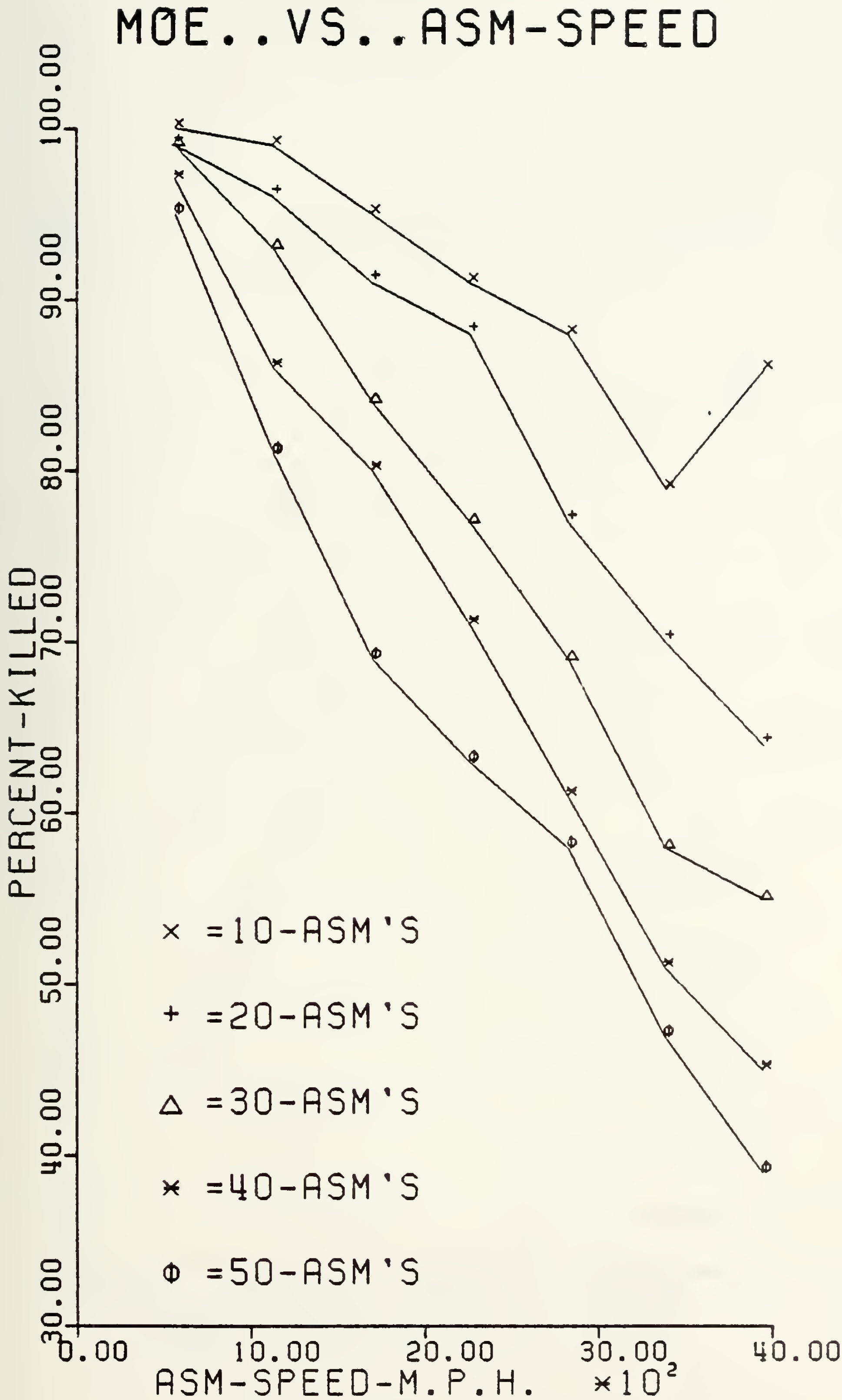
Figure 7

Figure 8 illustrates the MOE as a function of ASM speed. The graph shows that the MOE decreases with the increase in ASM speed. The results of the MOE as a function of defensive missile speed is illustrated in Figure 9. A significant increase in MOE is shown as defensive missile speed is increased to the speed of the ASM which is 2000 M.P.H. Once the defensive missile speed reaches the ASM speed, the increase in the MOE as a function of defensive missile speed is less significant. This result shows a high dependence of ASMD to the relative speed of the opposing missiles.

This dependence is also reflected in the result of the MOE as a function of ASM separation time as illustrated in Figure 10. A separation time of .20 minutes and an ASM speed of 2000 M.P.H. would be the separation time where only one ASM would be present in the playing area at a given time. When the separation time is shorter, more than one ASM will be in the playing area at a given time. The graph shows an increase in slope and then a flattening of the curve in the higher separation time.

The MOE as a function of assessment time or the MOE as a function of acquisition time showed little response as illustrated in Figures 11 and 12.

The number of scenarios which could be evaluated with SEAGUARD using a parametric analysis are limitless. The parametric study presented here is one example used to show the sensitivity of the model using a general scenario. It is by no means a thorough study of the model.



MØE..VS..ASM-SPEED
PERCENT-KILLED
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
0.00
10.00
20.00
30.00
40.00
ASM-SPEED-M.P.H. ×10²
× =10-ASM'S
+ =20-ASM'S
△ =30-ASM'S
✱ =40-ASM'S
Φ =50-ASM'S

Figure 8

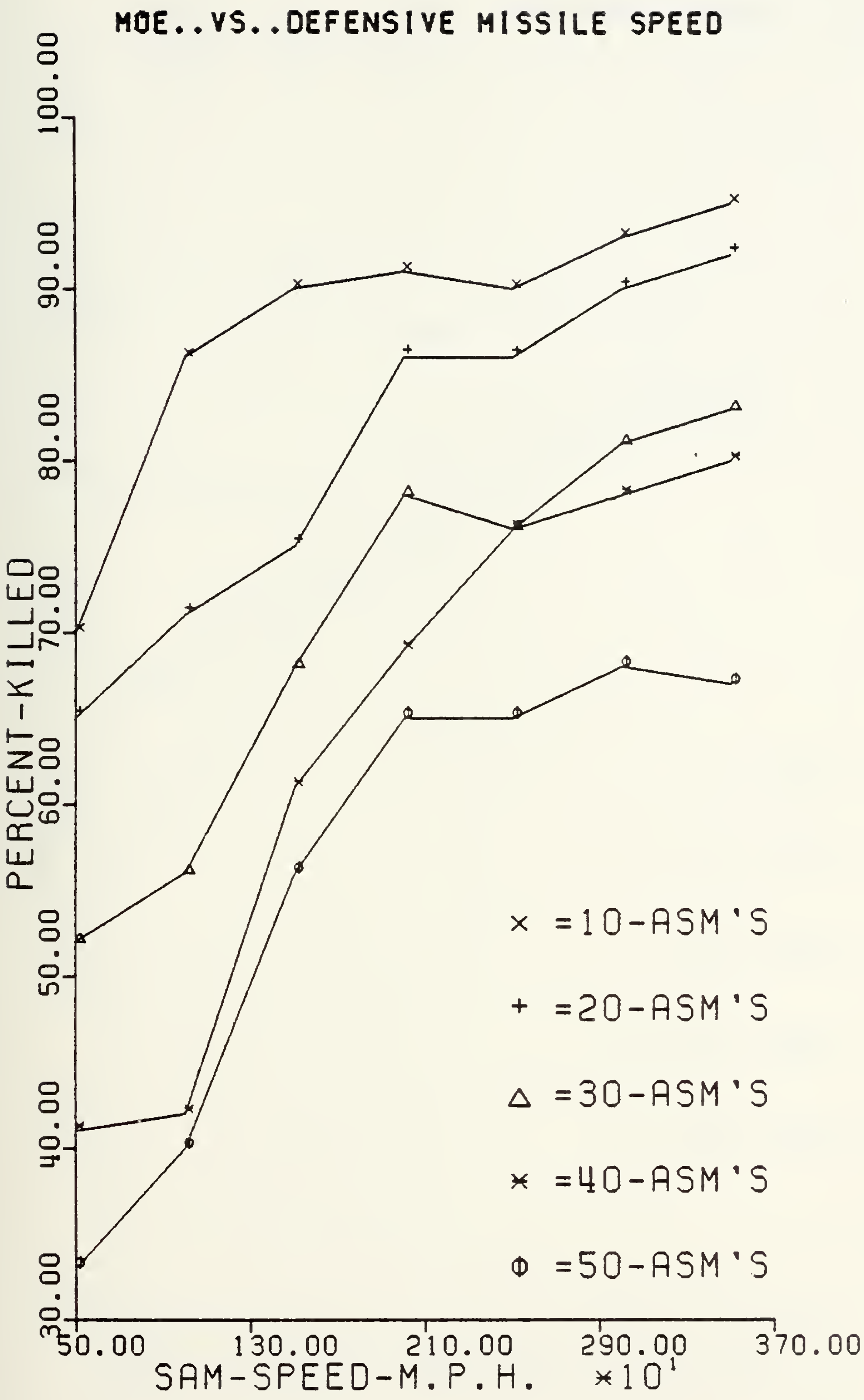

MOE..VS..DEFENSIVE MISSILE SPEED
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
PERCENT-KILLED
50.00
130.00
210.00
290.00
370.00
SAM-SPEED-M.P.H. ×10[1]
× =10-ASM'S
+ =20-ASM'S
△ =30-ASM'S
✶ =40-ASM'S
Φ =50-ASM'S

Figure 9

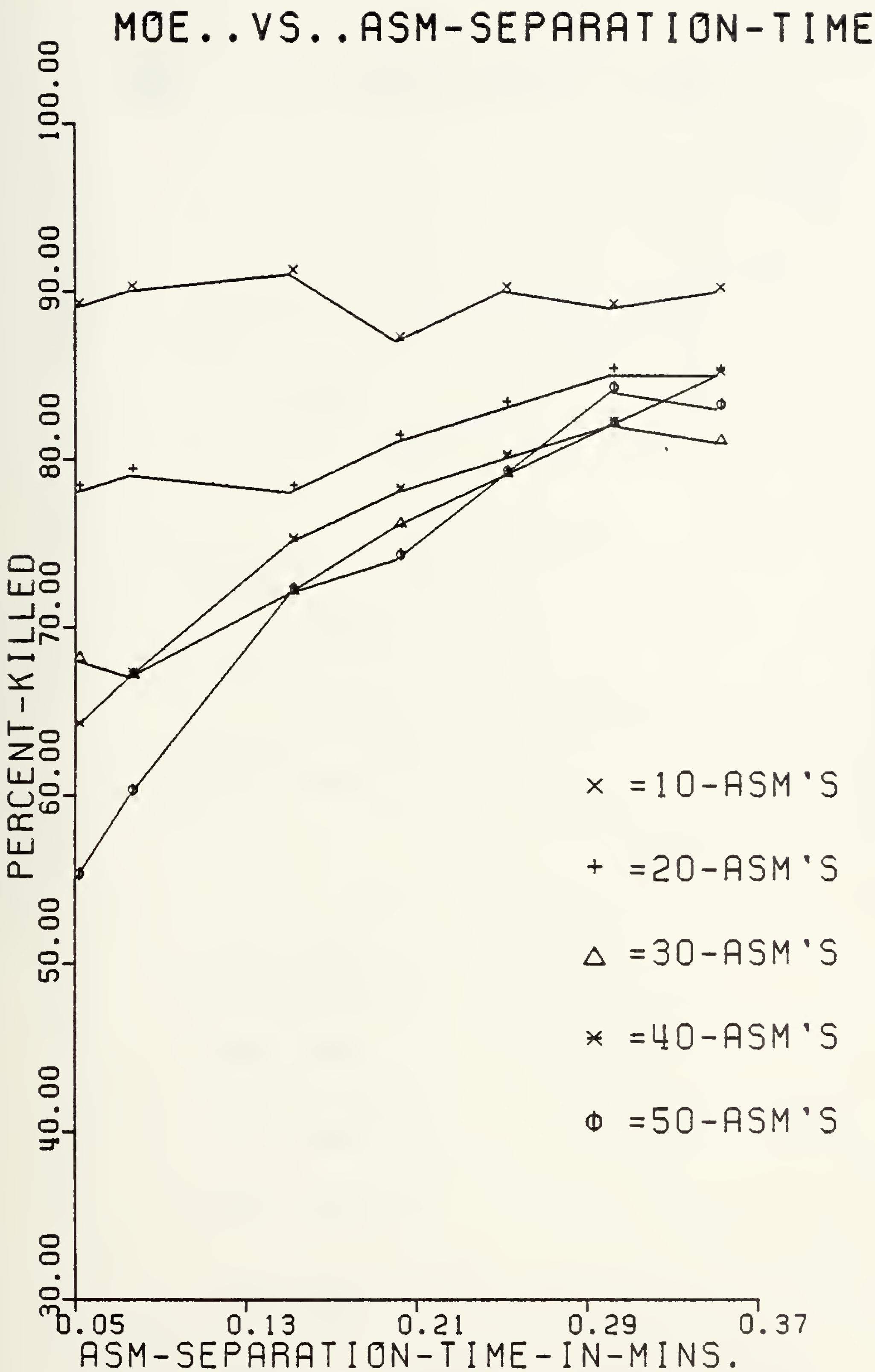

MOE..VS..ASM-SEPARATION-TIME
PERCENT-KILLED
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
0.05
0.13
0.21
0.29
0.37
ASM-SEPARATION-TIME-IN-MINS.
× =10-ASM'S
+ =20-ASM'S
△ =30-ASM'S
⋇ =40-ASM'S
Φ =50-ASM'S

Figure 10

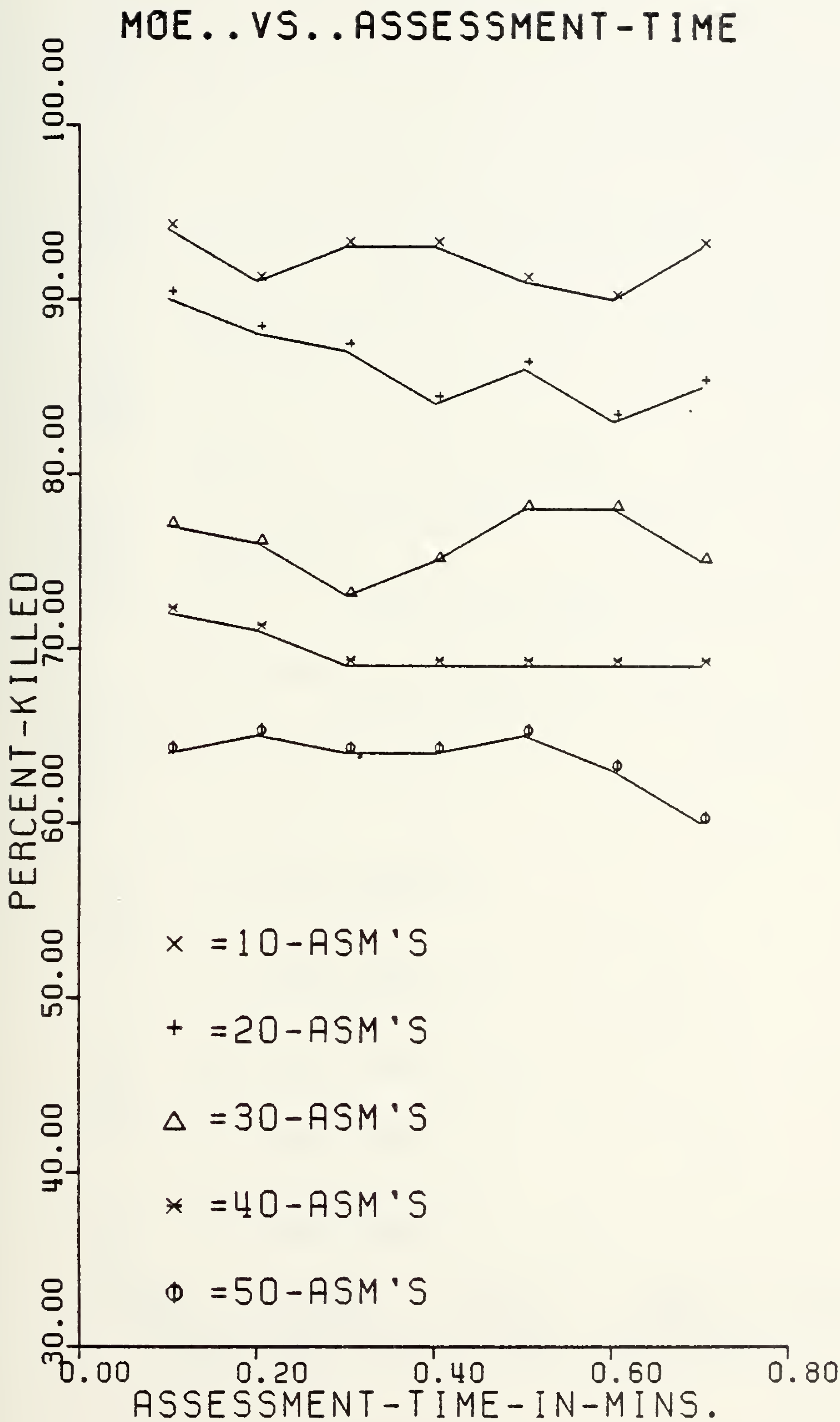
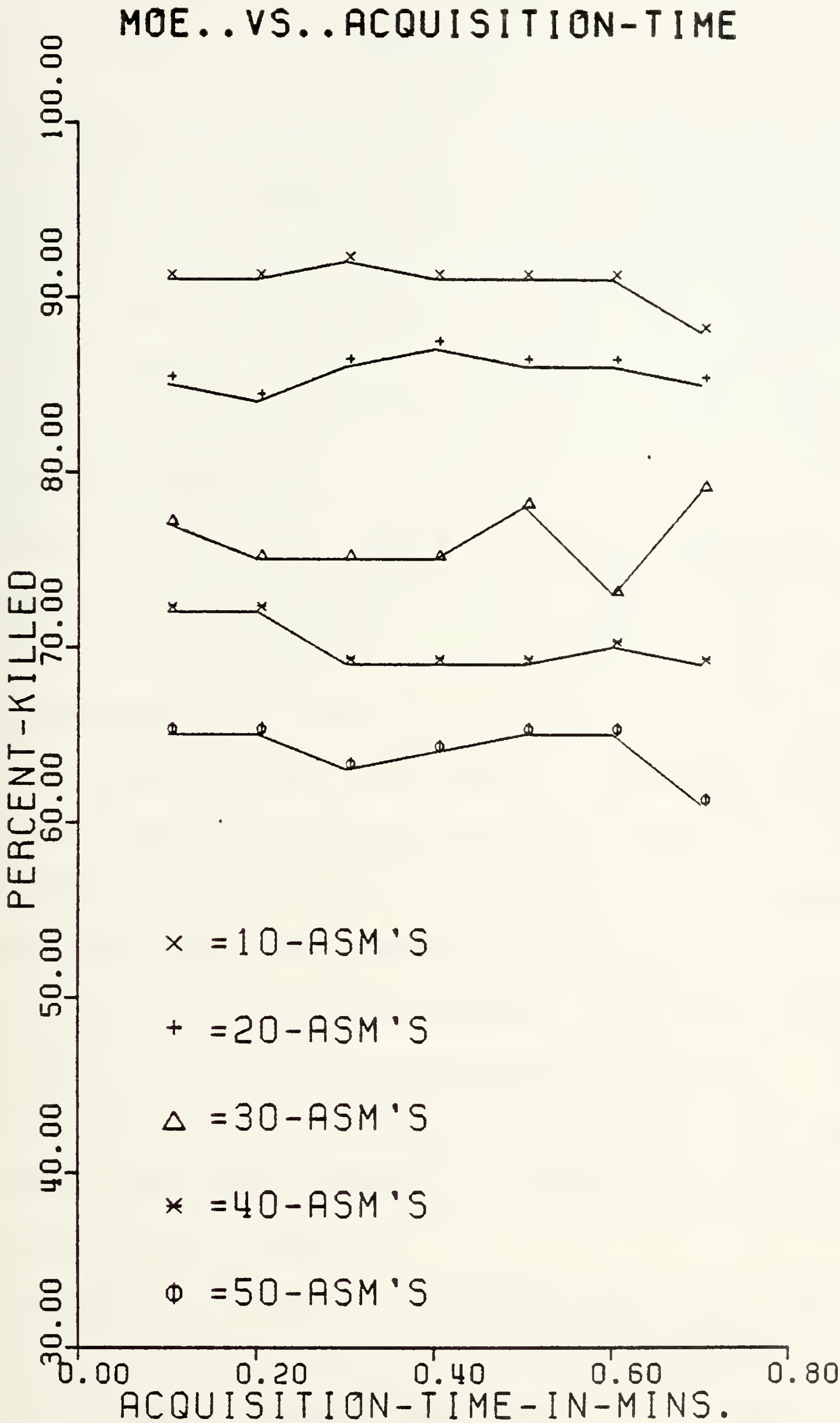

MOE..VS..ASSESSMENT-TIME
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
PERCENT-KILLED
× =10-ASM'S
+ =20-ASM'S
△ =30-ASM'S
⋇ =40-ASM'S
Φ =50-ASM'S
0.00
0.20
0.40
0.60
0.80
ASSESSMENT-TIME-IN-MINS.

Figure 11



MOE..VS..ACQUISITION-TIME
PERCENT-KILLED
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
0.00
0.20
0.40
0.60
0.80
ACQUISITION-TIME-IN-MINS.
× =10-ASM'S
+ =20-ASM'S
△ =30-ASM'S
⋇ =40-ASM'S
Φ =50-ASM'S

Figure 12

VI. CONCLUSIONS

The major concern driving this research has been the desire to develop a computer simulation which would effectively evaluate the ASMD capability of a naval force. A simulation having this characteristic has merit not only for use in force sizing, but also as an effective tool in evaluating tactics, variations in force disposition and relative platform efficiency.

In the course of this research, several conclusions have been reached. The evaluation of force size is highly dependent upon the models ability to play the real world elements of the encounters. In actuality, all the significant factors may not be known or cannot be modeled. The effectiveness of the model depends on its ability to be sensitive to changes of the important factors of the encounter. A very sophisticated model may take into account many elements of the encounter but in so doing may require a large detailed data base, large core size, and large computer run times. In addition, the increased amount of detail in the model may not yield a significant increase in sensitivity or increase the validity of the analysis.

The parametric analysis of SEAGUARD shows that the model responds to changes in those physical factors deemed to be important in ASMD. SEAGUARD can be used as an effective evaluator of relative differences in force size, force disposition and alternative tactics.

There are weaknesses in SEAGUARD in that only the uniform distribution was used for the generation of random numbers and the reliance on a symmetrical disposition of defensive forces to accomplish the effect of passing detection information. These weaknesses were accepted in order to maintain a relatively small scale code and short computer run time.

SEAGUARD should be evaluated further. A suggestion for further study is a comparative analysis between the output of SPEARS and the output of SEAGUARD for various scenarios. SEAGUARD could easily be expanded to include various distributions for use in the computation of random values.

APPENDIX A

VERBAL FLOW

SEAGUARD is comprised of the following major program segments:

(1) Initialization or set up

(2) Computation of play factors

(3) TNE Take Next Event

(4) SNE Schedule Next Event

(5) Reload event

(6) Tracking radar free event

(7) Intercept event

(8) Fire event

(9) Output

(10) Fire event

(11) Output

(12) Repeat and end

The initialization or set up segment reads the inputs, initializes all matrices and arrays, sets constants, and prints out the initial set up of the game. Once the set up is completed, ASM entry points, ASM speed, ASM altitude, radar horizon, distance to point of closest approach from the missile areas for the ASMs, time at point of closest approach, and the time of missile area entry are computed by the computation segment. During this computational segment, a fire event is generated by the SNE segment for each

ASM as it enters a new defensive missile area. After the computational segment completes execution it passes control to the TNE. The TNE then processes the schedule of events in chronological order. During the execution of these events either the event is completed, in which case control is then returned to TNE, or the event cannot be completed, in which case control is passed to SNE for a possible rescheduling of the event and then is passed back to TNE. Once the schedule of events has been exhausted or the maximum time of the simulation has been reached, control is transferred to the output segment. After output, control is transferred to the repeat and end segment which either repeats the process for further runs or ends execution of the program.

```
C                                              S E A G U A R D
C
C
      DIMENSION CALT(10), BTLF(50,20)
      DIMENSION IBA(50), IBMEC(50,10), BME(50,10), BVX(50,20), BVY(50,20
     1), APK(10), MTRF(10), IKILL(50), ITOT(50), MTRFO(10), MALO(10), AMS
     2(50,10,2), AMT(50,2), ABAR(50,2), AVAR(50,2), ASD(50,2), BM(10,2),
     3 BMT(2), BV(10,2), BVT(2), BSD(10,2), BSDT(2), PS(2), EVS(500), IES
     4VS1(500), IEVS2(500), IEVS3(500), AX(10), AY(10), MAL(10), RMR(10)
     5, RMAX(10), AVS(10), ATM(10), ATX(10), ASM(10), ASX(10), RTM(10)
      DIMENSION RTX(10), PK(10), BX(50,20), BY(50,20), BS(50,20), BT(50,
     120), BRH(50,20), ALT(50,20), BTMA(50,10,20), BTPC(50,10,20), BDPC(
     250,10,20), CTOTS(50,10,20), CTOT(50,10), CBAR(50,10), CVAR(50,2),
     3CSD(50,2), DM(10,2), DMT(2), DV(10,2), DVT(2), DSD(10,2), DSDT(2),
     4 IFE(50), AAA(50)
      INTEGER OUTPUT
C
C             READ INPUT
C
C
   20 DO 30 J=1,10
      READ (5,2010) MTRF(J),MAL(J),AX(J),AY(J),RMR(J),RMAX(J),AVS(J),PK(
     1J),ATM(J),ATX(J),ASM(J),ASX(J),RTM(J),RTX(J),CALT(J)
   30 CONTINUE
C
      READ (5,2020) BSX,BSM,BTX,BTM
      READ (5,2030) NMISAR,IEND,NBMAX,NBOMB,NBDEL,GX,GY,RCC,CPA,TMAX,NRP
     1L,AMAX,AMIN,STEP,NFLAG,IHIST,OUTPUT
C
C          SET CONSTANTS
C
      IF (NFLAG-1) 40,40,50
   40 IRUN = 0
      GO TO 60
   50 IRUN = 1
   60 NRRN = 0
      N = 1
      NREP = 0
      IEVS1(1) = 0
      IEVS2(1) = 0
      IEVS3(1) = 0
      IX = 13767
      IY = 13767
      YFL = 0.5
      YFL = RNG(IX,IY,YFL)
      NOL = 0
C
      DO 70 I=1,NBMAX
```

```
      IKILL(I) = 0
      IBA(I) = 0
      ITOT(I) = 0
      IFE(I) = 0
C
      DO 70 J=1,NMISAR
      BME(I,J) = 0.
      IBMEC(I,J) = 0
   70 CONTINUE
C
      IF (NMISAR-10) 80,100,1980
   80 L = NMISAR+1
C
      DO 90 K=1,2
C
      DO 90 J=L,10
      BM(J,K) = 0.
      BV(J,K) = 0.
      BSD(J,K) = 0.
      DM(J,K) = 0.
      DV(J,K) = 0.
      DSD(J,K) = 0.
C
      DO 90 I=1,NRPL
      AM(I,J,K) = 0.
      CTOTS(I,J,K) = 0.
   90 CONTINUE
C
C
C     PRINT INPUT
C
  100 CONTINUE
      WRITE (6,2040) GX,GY,RCC,CPA
C
      WRITE (6,2050)
      WRITE (6,2060)
      WRITE (6,2070) (AX(I),I=1,10)
      WRITE (6,2080) (AY(I),I=1,10)
      WRITE (6,2090) (CALT(I),I=1,10)
      WRITE (6,2100) (MAL(I),I=1,10)
      WRITE (6,2110) (MTRF(I),I=1,10)
      WRITE (6,2120) (RMR(I),I=1,10)
      WRITE (6,2130) (RMAX(I),I=1,10)
      WRITE (6,2140) (AVS(I),I=1,10)
      WRITE (6,2150) (ATM(I),I=1,10)
      WRITE (6,2160) (ATX(I),I=1,10)
      WRITE (6,2170) (ASM(I),I=1,10)
      WRITE (6,2180) (ASX(I),I=1,10)
```

```
      WRITE (6,2190) (RTM(I),I=1,10)
      WRITE (6,2200) (RTX(I),I=1,10)
C
      DO 110 I=1,10
  110 APK(I) = 100.*PK(I)
C
      WRITE (6,2210) (APK(I),I=1,10)
      WRITE (6,2220)
      WRITE (6,2230) BSX,BSM,AMAX,AMIN,BTX,BTM
C
C        INITIALIZE
C
C
      DO 120 I=1,NMISAR
      ATM(I) = ATM(I)/60.0
      ATX(I) = ATX(I)/60.0
      ASM(I) = ASM(I)/60.0
      ASX(I) = ASX(I)/60.0
      RTM(I) = RTM(I)/60.0
  120 RTX(I) = RTX(I)/60.0
C
      CPA = CPA/57.29577
      BTX = BTX/60.0
      BTM = BTM/60.0
      STEP = STEP /60.0
      NX = 13767
C
      DO 130 J=1,10
      MALO(J) = MAL(J)
  130 MTRFO(J) = MTRF(J)
C
C
C         COMPUTE  ASM   ENTRY POINTS
C
  140 RN = RNG(NX,IY,YFL)
      I = 1
  150 RN = RNG(IX,IY,YFL)
      B2 = 3.141592654
      B = 2.*B2-CPA*.5+CPA*RN
      IF (B.GT.(2.*B2)) B=B-(2.*B2)
      X = RCC*COS((.5*B2)-B)
      Y = RCC*SIN((.5*B2)-B)
      BX(I,N) = GX+X
      BY(I,N) = GY+Y
      RN = RNG(IX,IY,YFL)
C          COMPUTE  ASM   SPEED AT ENTRY
C
      BS(I,N) = (BSX-BSM)*RN+BSM
```

```
      BVX(I,N) = -BS(I,N)*COS((.5*B2)-B)
      BVY(I,N) = -BS(I,N)*SIN((.5*B2)-B)
C
C         COMPUTE  ASM   TIME AT ENTRY
C
      IF (I-1) 170,160,170
  160 BT(I,N) = 1.0
      GO TO 180
  170 RN = RNG(IX,IY,YFL)
      BT(I,N) = (BTX-BTM)*RN+BTM+BT(I-1,N)
C
C        COMPUTE  ASM   ALTITUDE AND BOMBER RADAR HORIZON
C
  180 AL = CALT(1)
C
      DO 190 J=1,NMISAR
      IF (CALT(J).GT.AL) AL=CALT(J)
  190 CONTINUE
C
      RN = RNG(IX,IY,YFL)
      ALT(I,N) = (AMAX-AMIN)*RN+AMIN
      BRH(I,N) = 1.25*SQRT((ALT(I,N)+AL)*1000.)
      J = 1
  200 A2 = (BX(I,N)-AX(J))**2+(BY(I,N)-AY(J))**2
      A = SQRT(A2)
      B2 = (AX(J)-GX)**2+(AY(J)-GY)**2
      B = SQRT(B2)
      C2 = RCC*RCC
      C = RCC
      CB = (-B2+A2+C2)/(2.*A*C)
      SB2 = 1.0-CB**2
      SB = SQRT(SB2)
C
C     COMPUTE DISTANCE TO PT OF CLOSEST APPROACH FROM MISSILE AREA
C
      BDPC(I,J,N) = A*SB
      BTLF(I,N) = ((RCC-BDPC(I,J,N))/BS(I,N))+BT(I,N)
      IF (A-(RCC+BDPC(I,J,N))) 210,210,410
  210 IF (BDPC(I,J,N)-RMAX(J)) 220,220,410
  220 IF (A-RMR(J)) 380,380,230
  230 IF (BDPC(I,J,N)-RMR(J)) 240,240,410
  240 IF (BRH(I,N)-RMR(J)) 270,250,250
  250 IF (RMR(J)-RMAX(J)) 260,260,290
  260 Z2 = RMR(J)**2-BDPC(I,J,N)**2
      GO TO 360
  270 IF (BRH(I,N)-BDPC(I,J,N)) 410,280,280
  280 IF (BRH(I,N)-RMAX(J)) 350,350,290
  290 X = A*CB
```

```
      BTPC(I,J,N) = X/BS(I,N)+BT(I,N)
      Z = BDPC(I,J,N)**2
      ZR2 = RMR(J)**2-Z
      ZM2 = RMAX(J)**2-Z
      ZB2 = BRH(I,N)**2-Z
      ZM = SQRT(ZM2)
      IF (ZR2-ZB2) 300,310,310
  300 ZR = SQRT(ZR2)
      GO TO 320
  310 ZR = SQRT(ZB2)
  320 TRMAX = (X-ZM)/BS(I,N)+BT(I,N)
      TZR = (X-ZR)/BS(I,N)+BT(I,N)
      Z = RMAX(J)**2+ALT(I,N)**2
      S1 = TRMAX-SQRT(Z)/AVS(J)
      RN = RNG(IX,IY,YFL)
      Z = (ATX(J)-ATM(J))*RN+ATM(J)
      IF ((S1-Z)-TZR) 340,330,330
  330 BTMA(I,J,N) = S1
      IF (TRMAX.GE.(RCC/BS(I,N)+.001+BT(I,N))) GO TO 410
      GO TO 400
  340 S1 = TZR+Z
      GO TO 330
  350 Z2 = BRH(I,N)**2-BDPC(I,J,N)**2
  360 Z = SQRT(Z2)
      X = A*CB
C
C      COMPUTE TIME AT CLOSEST APPROACH
C
      BTPC(I,J,N) = X/BS(I,N)+BT(I,N)
      Y = X-Z
C
C      TIME MISSILE AREA ENTRY
C
      BTMA(I,J,N) = BT(I,N)+Y/BS(I,N)
  370 RN = RNG(IX,IY,YFL)
      S1 = BTMA(I,J,N)+(ATX(J)-ATM(J))*RN+ATM(J)
      BTMA(I,J,N) = S1
      IF (BTMA(I,J,N).GE.(RCC/BS(I,N)+.001+BT(I,N))) GO TO 410
      GO TO 400
  380 IF (A-BRH(I,N)) 390,390,230
  390 BTMA(I,J,N) = BT(I,N)
      X = SQRT(A**2-BDPC(I,J,N)**2)
      BTPC(I,J,N) = X/BS(I,N)+BT(I,N)
      GO TO 370
  400 IS1 = 5
      IS2 = I
      IS3 = J
      IFE(N) = IFE(N)+1
```

```
      ASSIGN 420 TO NEXT
      GO TO 580
  410 BTMA(I,J,N) = 0.
  420 IF (J-NMISAR) 430,440,1980
  430 J = J+1
      GO TO 200
  440 IF (I-NBOMB) 450,460,1980
  450 I = I+1
      GO TO 150
  460 RN = RNG(IX,IY,YFL)
      GO TO 470
C
C          TNE
C
  470 IF (NOL) 1980,1200,480
  480 T = EVS(1)
      IE1 = IEVS1(1)
      IE2 = IEVS2(1)
      IE3 = IEVS3(1)
      IF (IHIST) 1980,500,490
  490 WRITE (6,2240) T, IE1, IE2, IE3
  500 IF (NOL-1) 1980,530,510
C
  510 DO 520 I=2,NOL
      EVS(I-1) = EVS(I)
      IEVS1(I-1) = IEVS1(I)
      IEVS2(I-1) = IEVS2(I)
  520 IEVS3(I-1) = IEVS3(I)
C
  530 EVS(NOL) = 0.
      IEVS1(NOL) = 0
      IEVS2(NOL) = 0
      IEVS3(NOL) = 0
      NOL = NOL-1
      IF (T-TMAX) 540,540,570
  540 IF (IE1) 1980,1980,550
  550 IF (IE1-5) 560,560,1980
  560 GO TO (720,740,750,900,910), IE1
  570 WRITE (6,2250) NBOMB,I
      GO TO 1200
C
C          SNE
C
  580 IF (S1.GT.BTLF(I,N).AND.IS1.EQ.5) GO TO 670
      IF (NOL) 1980,590,600
  590 IF (NOL.GT.500) GO TO 1980
      M = 1
      GO TO 640
```

```
600 M = NOL
610 IF (S1-EVS(M)) 620,690,680
620 EVS(M+1) = EVS(M)
    IEVS1(M+1) = IEVS1(M)
    IEVS2(M+1) = IEVS2(M)
    IEVS3(M+1) = IEVS3(M)
    IF (M-1) 1980,640,630
630 M = M-1
    GO TO 610
640 IF (IHIST) 1980,660,650
650 WRITE (6,2260) S1,IS1,IS2,IS3
660 EVS(M) = S1
    IEVS1(M) = IS1
    IEVS2(M) = IS2
    IEVS3(M) = IS3
    NOL = NOL+1
670 GO TO NEXT, (420,470,810,820,840,860,870,890,1170,1690)
680 M = M+1
    GO TO 640
690 IF (IS1-IEVS1(M)) 620,700,680
700 IF (IS2-IEVS2(M)) 620,710,680
710 IF (IS3-IEVS3(M)) 620,670,680
C
C     RELOAD
C
720 J = IE3
    IF (CALT(J).GT.0) GO TO 730
    MAL(J) = MAL(J)+1
    GO TO 470
730 MAL(J) = MALO(J)   .
    GO TO 470
C
C  T  R  FREE
C
740 J = IE3
    MTRF(J) = MTRF(J)+1
    GO TO 470
C
C    INTERCEPT
C
750 I = IE2
    J = IE3
    IF (IBA(I)) 470,760,470
760 RN = RNG(IX,IY,YFL)
    IF (RN-PK(J)) 770,770,850
770 IF (IRUN) 830,780,880
780 K = 1
790 IF (BME(I,K)) 800,820,800
```

```
800 RN = RNG(IX,IY,YFL)                                    S  03370
    S1 = T+(ASX(K)-ASM(K))*RN+ASM(K)                       S  03380
    BME(I,K) = S1                                          S  03390
    IS3 = K                                                S  03400
    IS1 = 2                                                S  03410
    IS2 = I                                                S  03420
    ASSIGN 810 TO NEXT                                     S  03430
    GO TO 580                                              S  03440
810 IS1 = 4                                                S  03450
    IS2 = I                                                S  03460
    ASSIGN 820 TO NEXT                                     S  03470
    GO TO 580                                              S  03480
820 IF (K-NMISAR) 830,840,830                              S  03490
830 K = K+1                                                S  03500
    GO TO 790                                              S  03510
840 IBA(I) = 1                                             S  03520
    IKILL(I) = J                                           S  03530
    GO TO 470                                              S  03540
850 RN = RNG(IX,IY,YFL)                                    S  03550
    S1 = T+(ASX(J)-ASM(J))*RN+ASM(J)                       S  03560
    BME(I,J) = S1                                          S  03570
    IS1 = 5                                                S  03580
    IS2 = I                                                S  03590
    IS3 = J                                                S  03600
    ASSIGN 860 TO NEXT                                     S  03610
    GO TO 580                                              S  03620
860 IS1 = 2                                                S  03630
    IS2 = I                                                S  03640
    ASSIGN 870 TO NEXT                                     S  03650
    GO TO 580                                              S  03660
870 IS1 = 4                                                S  03670
    IS2 = I                                                S  03680
    ASSIGN 470 TO NEXT                                     S  03690
    GO TO 580                                              S  03700
880 RN = RNG(IX,IY,YFL)                                    S  03710
    S1 = T+(ASX(J)-ASM(J))*RN+ASM(J)                       S  03720
    BME(I,J) = S1                                          S  03730
    IS1 = 2                                                S  03740
    IS2 = I                                                S  03750
    IS3 = J                                                S  03760
    ASSIGN 890 TO NEXT                                     S  03770
    GO TO 580                                              S  03780
890 IS1 = 4                                                S  03790
    ASSIGN 840 TO NEXT                                     S  03800
    GO TO 580                                              S  03810
C                                                          S  03820
C   ENGAGEMENT                                             S  03830
C                                                          S  03840
```

```
  900 I = IE2
      J = IE3
      BME(I,J) = 0.
      GO TO 470
C
C     FIRE EVENT
C
  910 I = IE2
      J = IE3
      IF (T-BTPC(I,J,N)) 920,470,470
  920 IF (IBA(I)) 470,930,470
  930 IF (MAL(J)) 940,1010,940
  940 IF (MTRF(J)) 950,1030,950
  950 IF (IRUN) 960,1040,960
  960 K = 1
  970 IF (BME(I,K)) 980,990,980
  980 S1 = BME(I,K)
      GO TO 1020
  990 IF (K-NMISAR) 1000,1040,1000
 1000 K = K+1
      GO TO 970
 1010 S1 = T+STEP
 1020 IS1 = 5
      IS2 = I
      IS3 = J
      ASSIGN 470 TO NEXT
      GO TO 580
 1030 S1 = T+STEP
      GO TO 1020
 1040 XA = BX(I,N)+BVX(I,N)*(T-BT(I,N))
      YA = BY(I,N)+BVY(I,N)*(T-BT(I,N))
      A = BS(I,N)**2-AVS(J)**2
      B = 2.0*(BVX(I,N)*(XA-AX(J))+BVY(I,N)*(YA-AY(J)))
      CC = ((ALT(I,N)-CALT(J))**2)
      IF (CALT(J).GT.ALT(I,N)) CC=((CALT(J)-ALT(I,N))*1000./5280.)**2
      C = (XA-AX(J))**2+(YA-AY(J))**2+CC
      IF (A) 1050,1140,1050
 1050 Z = B**2-4.0*A*C
      IF (Z) 470,1060,1060
 1060 TF1 = (-B+SQRT(Z))/(2.0*A)
      TF2 = (-B-SQRT(Z))/(2.0*A)
      IF (TF1) 1070,470,1070
 1070 IF (TF2) 1080,470,1080
 1080 IF (TF1) 1110,1090,1090
 1090 IF (TF2) 1100,1130,1130
 1100 TF = TF1
      GO TO 1150
 1110 IF (TF2) 470,1120,1120
```

```
1120 TF = TF2
     GO TO 1150
1130 IF (TF1-TF2) 1100,1120,1120
114C TF = -C/B
     IF (TF) 470,470,1150
1150 R = (TF*AVS(J))**2-CC
     R = SQRT(R)
     IF ((R-1.)-RMAX(J)) 1160,1160,119C
1160 S1 = T+TF
     IS1 = 3
     IS2 = I
     IS3 = J
     ASSIGN 1170 TO NEXT
     GO TO 580
1170 MAL(J) = MAL(J)-1
     MTRF(J) = MTRF(J)-1
     IBMEC(I,J) = IBMEC(I,J)+1
     BME(I,J) = S1
     IF (MAL(J).GT.0.AND.CALT(J).GT.0.C) ASSIGN 470 TO NEXT
     IF (MAL(J).GT.0.AND.CALT(J).GT.0.0) GO TO 470
     RN = RNG(IX,IY,YFL)
     S1 = T+(RTX(J)-RTM(J))*RN+RTM(J)
     IS1 = 1
     IS2 = I
     IS3 = J
1180 ASSIGN 470 TO NEXT
     GO TO 580
1190 S1 = T+STEP
     IS1 = 5
     IS2 = I
     IS3 = J
     GO TO 1180
C
C    OLTPUT
C
1200 IF (OUTPUT) 1980,1210,1300
C
1210 DO 1220 I=1,NBOMB
     ITOT(I) = 0
C
     DO 1220 J=1,NMISAR
1220 ITOT(I) = ITOT(I)+IBMEC(I,J)
C
     IF (NFLAG-1) 1230,1240,1250
1230 IF (IRUN) 1980,1240,1250
1240 WRITE (6,2270) N
     GO TO 1260
1250 WRITE (6,2280) N
```

```
 1260 WRITE (6,2290)
C
      DO 1270 I=1,NBOMB
 1270 WRITE (6,2300) I,BX(I,N),BY(I,N),BRH(I,N),ALT(I,N),BS(I,N)
C
      WRITE (6,2310)
C
      DC 1280 I=1,NBOMB
 1280 WRITE (6,2320) I,BT(I,N),(BTMA(I,J,N),J=1,NMISAR)
C
      WRITE (6,2330)
C
      DO 1290 I=1,NBOMB
 1290 WRITE (6,2340) I,(IBMEC(I,J),J=1,NMISAR),ITOT(I),IKILL(I)
C
      WRITE (6,2350)
      GO TO 1300
 1300 I = N
      IF (IRUN) 1980,1310,1320
 1310 K = 1
      GC TO 1330
 1320 K = 2
C
 1330 DO 1340 J=1,NMISAR
      CTOTS(I,J,K) = 0
      AM(I,J,K) = 0
C
      DO 1340 L=1,NBOMB
      B = IBMEC(L,J)
      CTOTS(I,J,K) = CTOTS(I,J,K)+B
 1340 CONTINUE
C
C
      DO 1360 L=1,NBOMB
      IF (IKILL(L)) 1980,1360,1350
 1350 J = IKILL(L)
      AM(I,J,K) = AM(I,J,K)+1.0
 1360 CONTINUE
C
      CTOT(I,K) = 0.
      AMT(I,K) = 0.
C
      DO 1370 J=1,NMISAR
      CTOT(I,K) = CTOT(I,K)+CTOTS(I,J,K)
 1370 AMT(I,K) = AMT(I,K)+AM(I,J,K)
C
      A = NMISAR
      ABAR(I,K) = AMT(I,K)/A
```

```
      CBAR(I,K) = CTOT(I,K)/A
      CVAR(I,K) = 0.
      AVAR(I,K) = 0.
C
      DO 1380 J=1,NMISAR
      CVAR(I,K) = CVAR(I,K)+(CTOTS(I,J,K)-CBAR(I,K))**2
 1380 AVAR(I,K) = AVAR(I,K)+(AM(I,J,K)-ABAR(I,K))**2
C
      AVAR(I,K) = AVAR(I,K)/A
      CVAR(I,K) = CVAR(I,K)/A
      CSD(I,K) = SQRT(CVAR(I,K))
      ASD(I,K) = SQRT(AVAR(I,K))
      IF (N-NRPL) 1620,1390,1620
 1390 IF (NFLAG-1) 1400,1410,1540
 1400 IF (IRUN) 1410,1620,1410
 1410 K = 1
 1420 A = NRPL
C
      DO 1440 J=1,NMISAR
      DM(J,K)=0.
      BM(J,K)=0.
C
      DO 1430 I=1,NRPL
      DM(J,K) = DM(J,K)+CTOTS(I,J,K)
 1430 BM(J,K) = BM(J,K)+AM(I,J,K)
C
      DM(J,K) = DM(J,K)/A
 1440 BM(J,K) = BM(J,K)/A
C
      DMT(K)=0.
      BMT(K)=0.
C
      DO 1450 I=1,NRPL
      DMT(K) = DMT(K)+CTOT(I,K)
 1450 BMT(K) = BMT(K)+AMT(I,K)
C
      DMT(K) = DMT(K)/A
      BMT(K) = BMT(K)/A
C
      DO 1470 J=1,NMISAR
      DV(J,K)=0.
      BV(J,K)=0.
C
      DO 1460 I=1,NRPL
      DV(J,K) = DV(J,K)+(CTOTS(I,J,K)-DM(J,K))**2
 1460 BV(J,K) = BV(J,K)+(AM(I,J,K)-BM(J,K))**2
C
      DV(J,K) = DV(J,K)/A
```

```
 1470 BV(J,K)  =  BV(J,K)/A
C
      DVT(K)=0.
      BVT(K)=0.
C
      DO 1480 I=1,NRPL
      DVT(K)  =  DVT(K)+(CTOT(I,K)-DMT(K))**2
 1480 BVT(K)  =  BVT(K)+(AMT(I,K)-BMT(K))**2
C
      DVT(K)  =  DVT(K)/A
      BVT(K)  =  BVT(K)/A
C
      DO 1490 J=1,NMISAR
      DSD(J,K)  =  SQRT(DV(J,K))
 1490 BSD(J,K)  =  SQRT(BV(J,K))
C
      DSDT(K)  =  SQRT(DVT(K))
      BSDT(K)  =  SQRT(BVT(K))
      AAA(K)  =  0.
      AA  =  0.
      A  =  NBOMB
      PS(K)  =  0.
C
      DO 1500 I=1,NRPL
      AA  =  AA+AMT(I,K)
 1500 CONTINUE
C
      AAA(K)  =  AA/A
C
      DO 1520 I=1,NRPL
      IF (AMT(I,K)-A) 1520,1510,1520
 1510 PS(K)  =  PS(K)+1.0
 1520 CONTINUE
C
      A  =  NRPL
      PS(K)  =  PS(K)/A
      AAA(K)  =  AAA(K)/A
      IF (NFLAG-1) 1530,1550,1610
 1530 IF (K-2) 1540,1550,1540
 1540 K  =  2
      GO TO 1420
 1550 K  =  1
      WRITE (6,2360)
      GO TO 1590
 1560 IF (NFLAG-1) 1580,1580,1570
 1570 WRITE (6,2370)
 1580 WRITE (6,2380)
 1590 WRITE (6,2390)
```

```
      WRITE (6,2400)
      WRITE (6,2410) (NBOMB,I,(CTOTS(I,J,K),J=1,10),CTOT(I,K),CBAR(I,K),
     1CVAR(I,K),CSD(I,K),I=1,NRPL)
      WRITE (6,2350)
      WRITE (6,2420)
      WRITE (6,2430) (DM(J,K),J=1,10),DMT(K)
      WRITE (6,2440) (DV(J,K),J=1,10),DVT(K)
      WRITE (6,2450) (DSD(J,K),J=1,10),DSDT(K)
      WRITE (6,2460)
      WRITE (6,2400)
      WRITE (6,2410) (NBOMB,I,(AM(I,J,K),J=1,10),AMT(I,K),ABAR(I,K),AVAR
     1(I,K),ASD(I,K),I=1,NRPL)
      WRITE (6,2350)
      WRITE (6,2420)
      WRITE (6,2430) (BM(J,K),J=1,10),BMT(K)
      WRITE (6,2440) (BV(J,K),J=1,10),BVT(K)
      WRITE (6,2450) (BSD(J,K),J=1,10),BSDT(K)
      WRITE (6,2470) PS(K)
      WRITE (6,2480) AAA(K)
      IF (NFLAG-1) 1600,1620,1620
 1600 IF (K-2) 1610,1620,1980
 1610 K = 2
      GO TO 1560
C
C     REPEAT AND END
C
 1620 NOL = 0
      IF (NFLAG-1) 1630,1780,1780
 1630 IF (IRUN) 1980,1640,1780
 1640 IRUN = 1
      ASSIGN 1740 TO LAST
 1650 I = 1
 1660 J = 1
 1670 IF (BTMA(I,J,N)) 1680,1690,1680
 1680 S1 = BTMA(I,J,N)
      IS1 = 5
      IS2 = I
      IS3 = J
      ASSIGN 1690 TO NEXT
      GO TO 580
 1690 IF (J-NMISAR) 1700,1710,1980
 1700 J = J+1
      GO TO 1670
 1710 IF (I-NBOMB) 1720,1730,1980
 1720 I = I+1
      GO TO 1660
 1730 GO TO LAST, (1740,1880,1950)
C
```

```
 1740 DO 1750 I=1,NBOMB
      IBA(I) = 0
      IKILL(I) = 0
      ITOT(I) = 0
C
      DO 1750 J=1,NMISAR
      IBMEC(I,J) = 0
      BME(I,J)=0.
 1750 CONTINUE
C
C
      DO 1760 J=1,NMISAR
      MAL(J) = MALO(J)
 1760 MTRF(J) = MTRFO(J)
C
      RN = RNG(IX,IY,YFL)
      IF (NFLAG-1) 1770,1770,1960
 1770 IF (IRUN) 1980,1960,470
 1780 IF (N-NRPL) 1790,1890,1980
 1790 IF (NREP) 1980,1800,1840
 1800 IF (NFLAG-1) 1820,1820,1810
 1810 IRUN = 1
      GO TO 1830
 1820 IRUN = 0
 1830 N = N+1
      GO TO 1740
 1840 IF (NFLAG-1) 1860,1860,1850
 1850 IRUN = 1
      GO TO 1870
 1860 IRUN = 0
 1870 N = N+1
      NBOMB = NBOMB-NBDEL
      ASSIGN 1880 TO LAST
      GO TO 1650
 1880 GO TO 1950
 1890 IF (NBOMB-NBMAX) 1900,1990,1990
 1900 NBOMB = NBOMB+NBDEL
      IF (NBOMB-NBMAX) 1910,1910,1990
 1910 NBOMB = NBOMB-NBDEL
      IF (NFLAG-1) 1930,1930,1920
 1920 IRUN = 1
      GO TO 1940
 1930 IRUN = 0
 1940 N = 1
      NREP = 1
      ASSIGN 1950 TO LAST
      GO TO 1650
 1950 NBOMB = NBOMB+NBDEL
```

```
      GO TO 1740
 1960 IF (NREP) 1980,140,1970
 1970 I = NBOMB-NBDEL+1
      RN = RNG(NX,IY,YFL)
      GO TO 150
 1980 WRITE (6,2490)
 1990 IF (IEND) 2000,2000,20
 2000 CONTINUE
      STOP
C
 2010 FORMAT (I1,I1,F6.0,F6.0,2F3.0,F5.0,8F4.0)
 2020 FORMAT (2F5.0,2F4.0)
 2030 FORMAT (I2,I1,I2,I2,I2,3F6.0,F5.0,F2.0,I2,3F4.0,3I1)
 2040 FORMAT (5X,15HGAME PARAMETERS//,10X,30H XCOORDINATE OF CENTER (GX)
     1 = F8.2,2X,5HMILES/10X,30H YCOORDINATE OF CENTER (GY) = F8.2,2X,5H
     2MILES/10X,31H RADIUS OF CIRCLE        (RCC) = F7.2,2X,5HMILES/10X,31
     3H CENTRAL PLAYING ANGLE (CPA) = F7.2,2X,7HDEGREES////)
 2050 FORMAT (25H MISSILE AREA INFORMATION//1X,16H MISSILE AREA NR)
 2060 FORMAT (9X,1H1,9X,1H2,9X,1H3,9X,1H4,9X,1H5,9X,1H6,9X,1H7,9X,1H8,9X
     1,1H9,9X,2H10,/)
 2070 FORMAT (7X,10(F6.2,4X),5HMILES,7X,2HAX)
 2080 FORMAT (7X,10(F6.2,4X),5HMILES,7X,2HAY)
 2090 FORMAT (7X,10(F6.2,4X),5HK FT ,7X,4HCALT)
 2100 FORMAT (10(7X,I3),19X,3HMAL)
 2110 FORMAT (10(7X,I3),19X,4HMTRF)
 2120 FORMAT (7X,10(F6.2,4X),5HMILES,7X,3HRMR)
 2130 FORMAT (7X,10(F6.2,4X),5HMILES,7X,4HRMAX)
 2140 FORMAT (6X,10(F7.2,3X),1X,5HMI/HR,7X,3HAVS)
 2150 FORMAT (7X,10(F6.2,4X),5HMINS ,7X,3HATM)
 2160 FORMAT (7X,10(F6.2,4X),5HMINS ,7X,3HATX)
 2170 FORMAT (7X,10(F6.2,4X),5HMINS ,7X,3HASM)
 2180 FORMAT (7X,10(F6.2,4X),5HMINS ,7X,3HASX)
 2190 FORMAT (7X,10(F6.2,4X),5HMINS ,7X,3HRTM)
 2200 FORMAT (7X,10(F6.2,4X),5HMINS ,7X,3HRTX)
 2210 FORMAT (7X,10(F6.2,4X),5HPER. ,7X,2HPK)
 2220 FORMAT (///19H  ASM   INFORMATION//)
 2230 FORMAT (5X,21H MAXIMUM SPEED (BSX)=F6.1,2X,8HMILES/HR/5X,21H MINIM
     1UM SPEED (BSM)=F6.1,2X,8HMILES/HR/5X,26H MAXIMUM ALTITUDE (AMAX)=
     2F8.1,2X,7H K FEET/5X,26H MINIMUM ALTITUDE (AMIN)= F8.1,2X,7H K FEE
     3T/5X,31H MAXIMUM SEPERATION TIME (BTX)=F7.2,2X,5HMINS./5X,31H MINI
     4MUM SEPERATION TIME (BTM)=F7.2,2X,5HMINS./)
 2240 FORMAT (10X,F7.3,3I5)
 2250 FORMAT (26HMAXIMUM TIME EXCEEDED FOR I5,8H ASM  S ,4HRUN I3)
 2260 FORMAT (20X,F7.3,3I5)
 2270 FORMAT (/////,35H UNCOORDINATED ATTACK RESULTS   REP,I3,//)
 2280 FORMAT (/////,33H COORDINATED ATTACK RESULTS   REP,I3,//)
 2290 FORMAT ('  ASM  XCCORD  YCOORD   ASMH    MALT  ASM SPEED',/)
 2300 FORMAT (I5,3F8.2,F7.2,F10.2)
```

```
2310 FORMAT (//,' ASM      TIME  TIME(1)  TIME(2)  TIME(3)  TIME(4)  TI
    1ME(5)  TIME(6)  TIME(7)  TIME(8)  TIME(9)  TIME(10)',/)
2320 FORMAT (I3,F9.3,10F9.3)
2330 FORMAT (//,' ASM  SMA1  SMA2  SMA3  SMA4  SMA5  SMA6  SMA7  SMA8
    1 SMA9  SMA10  TOTAL  A/K ',/)
2340 FORMAT (I4,9I6,I7,I7,I6)
2350 FORMAT (///)
2360 FORMAT (' SUMMARY UNCOORDINATED RESULT',//)
2370 FORMAT (1H1)
2380 FORMAT (' SUMMARY COORDINATED RESULTS',//)
2390 FORMAT (20X,'  ASM S  FIRED  ',//)
2400 FORMAT ('  ASM S  REP  MA1  MA2  MA3  MA4  MA5  MA6  MA7  MA8  MA9
    1  MA10   TOTAL   MEAN   VAR   SD',//)
2410 FORMAT (I7,I5,9F5.0,F6.0,F8.2,3F6.2)
2420 FORMAT (9X,'  MA1    MA2    MA3    MA4    MA5    MA6    MA7    MA8
    1     MA9    MA10    TOTAL',/)
2430 FORMAT (' MEAN  ',10F7.2,F9.2)
2440 FORMAT (' VAR   ',10F7.2,F8.2)
2450 FORMAT ('  SD   ',10F7.2,F8.2,////)
2460 FORMAT (20X,'  ASM  KILLS  ',//)
2470 FORMAT (///,27H PROBABILITY OF ZERO HITS =F7.2,/)
2480 FORMAT (/,26H PERCENT ASM'S KILLED  =  F7.2,/)
2490 FORMAT (6H ERROR)
     END
     REAL FUNCTIONRNG(IX,IY,YFL)
     IY = IX*65549
     IF (IY) 20,30,30
  20 IY = IY+2147483647+1
  30 YFL = IY
     YFL = YFL*.4656613E-9
     RNG = YFL
     IX = IY
     RETURN
     END
     END
```

BIBLIOGRAPHY

1. Catalog of Navy War Game Models, Department of the Navy, November 1967.

2. Catalog of War Gaming and Military Simulation Models, 7th ed., Studies, Analysis, and Gaming Agency; Organization, Joint Chiefs of Staff, August 1977.

3. Degroot, M.H., Probability and Statistics, Addison-Wesley, 1975.

4. Fishman, G.S., Concepts and Methods in Discrete Event Digital Simulation, Wiley, 1973.

5. Gaver, D.P. and Thompson, G.L., Programming and Probability Models in Operations Research, Brooks/Cole, 1973.

6. Naval Research Laboratory Report 7958, SPEARS, An AAW Performance Simulation, by D.J. Kaplan (NRL), L.C. Davis (NRL), M.E.B. Owens (NRL), O.F. Forsyth (SRI), and J.J. Penick (SRI), June 9, 1976.

7. U.S. Naval Postgraduate School Technical Report 67, A Computer Simulation for the Evaluation of Surface-to-Air Missile Systems in a Clear Environment, by A.F. Andrus, June 1966.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4.	LT Ellen F. Roland, USN, Code 55Ro Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
5.	Professor A.F. Andrus, Code 55As Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
6.	LT Alexander J. Callahan, Jr., USN 36 N. Stoughton St. Bergenfield, New Jersey 07621	1
7.	LCOL Edward P. Kelleher, USA, Code 55Ka Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
8.	SAGA, OJCS Room 1D936 Pentagon Washington, D.C. 20301	1
9.	LCDR J.S. Redd, USN CNO Executive Panel Room 405 1401 Wilson Blvd. Arlington, Virginia 22209	1

Thesis 180770
C1886 Callahan
c.1 Seaguard: a general purpose antiship missile defense computer simulation.